SPARSE AND LARGE-SCALE LEARNING MODELS AND ALGORITHMS FOR MINING HETEROGENEOUS BIG DATA

by

XIAO CAI

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2013

I dedicate this work to my parents...

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor Dr. Heng Huang for everything he has done during the four and half year of my doctoral studies in UT Arlington, especially for his generous support, patient guidance and invaluable inspiration. The dissertation would not be possible without him. It is Dr. Huang who introduced me to field of machine learning. His insight and experience have guided me through my research. His enthusiasm towards work has also influenced me a lot in my life. I have been very lucky to work with him.

Deep gratitude also goes to my committee members. I would like to thank my co-advisor Dr. Farhad Kamangar for providing me valuable comments and constructive advice of different image descriptors and MATLAB graphic user interphase, which provides the input source and makes the data visualization work much easier for the learning algorithms. I also want to express my sincere appreciation to Dr. Chris Ding for his great help and valuable discussions on low-rank linear regression, linear discriminant analysis and other simple but very important machine learning basis knowledge. His sharp critical thinking as well as mathematical rigor refines this dissertation. I am also grateful to Dr. Jeff Lei for serving as my committee member and taking time to criticize, proofread and improve the quality of this dissertation.

During my four and half years Ph.D. study, I interacted with many colleagues and I benefited a lot from the valuable discussions on research in the Computational Science Lab (CSL). Especially, I want to thank Dr. Feiping Nie, with whom I worked closely and puzzled over many interesting and changeling optimization problems. I also want to thank the former lab members, Dr. Nha Nguyen, Dr. Dijun Luo and Dr. Hua Wang for their knowledge sharing during my earlier Ph.D study. The work of current members in

CSL makes my life easier every day as well. It is my pleasure to work with Miao Zhang, Deguang Kong, De Wang and Xiaoqian Wang. I am very fortunate to have a lot of friends standing by my side in any situation. The friendship and scholarship are going to become beautiful memories in my life.

I would also like to extend my appreciation to my research intern mentors in Abbot Laboratories, Dr. Zhizhou Wang and Dr. Min Xie, who taught me how to transfer the knowledge leaned from the school to the real product and provided me with a great vision on practical applications of my research area.

The last but not the least, there are two persons in my life holding faith in me, and giving me endless love. They are my wonderful parents Hebei Cai and Weili Sheng. I would like to gratefully and sincerely thank them for all the support they have provided me over the years.

November 15, 2013

ABSTRACT

SPARSE AND LARGE-SCALE LEARNING MODELS AND ALGORITHMS FOR MINING HETEROGENEOUS BIG DATA

XIAO CAI, Ph.D.

The University of Texas at Arlington, 2013

Supervising Professor: Heng Huang

With the development of PC, internet as well as mobile devices, we are facing a data exploding era. On one hand, more and more features can be collected to describe the data, making the size of the data descriptor larger and larger. On the other hand, the number of data itself explodes and can be collected from multiple resources. When the data becomes large scale, the traditional data analysis method may fail, suffering the curse of dimensionality and etc. In order to explore and analyze the large-scale data more accurately and more efficiently, based on the characteristic of the data, we propose several learning algorithms to mine the Heterogeneous data. To be specific, if the feature dimension is large, we propose several sparse learning based feature selection methods to select the key words from the text or to find the bio-marker from the gene expression data; if the number of data itself is huge, we proposed multi-view K-Means method to do the clustering to avoid the heavy graph construction burden; if the data is represented or collected by multiple resources, we propose graph based multi-modality model to do semi-supervised learning and clustering. In addition, if the number of classes is large, we provides a global solution to the low-rank regression and proves that the low-rank regression is equivalent to doing

linear regression in LDA space. We empirically evaluate each of our proposed models on several benchmark data sets and our methods can consistently achieve superior results with the comparison of state-of-art methods.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Introduction

With the advent of modern high technologies, like PC, internet as well as mobile devices, we are drowning in a sea of data. Much of the data we regularly encounter nowadays is (1) high dimensional, i.e. the data descriptor has a large number of variables (features), possibly much more than observations (data samples); (2) the number of data samples is huge and can be represented or collected by multiple resources; (3) the number of classes is huge and there are some correlation between different classes. Blindly fitting traditional models to such data is prone to giving over-fitted or useless models with heavy computation burden posing great difficulty for further data analysis.

For the high dimensional data, although we can employ conventional dimension reduction method to reduce the number of features, for example PCA, LDA, and so on [1], we cannot tackle the problems where the features have natural meanings and they cannot be projected, such as text mining [2], DNA microarray [3], and mass spectrometry [4]. Therefore, feature selection, the process of selecting a subset of meaningful features, is a key issue in building robust data mining models for later classification, clustering, and other data analysis tasks since it can select text key words, discover biomarkers, speed up the learning process, boost the model generalization capability and alleviate the effect of *the curse of dimensionality* [5].

In addition, data can be collected from numerous resources or represented by many representations. In image segmentation, an image can be represented by many different visual descriptors. In web grouping, a web can be characterized by its content and anchor

texts of inbound hyperlink. In social network community discovery, researchers discover the hidden grouping relation (e.g. friend or knows) in the network via personal interest or geographic information. In text mining, people study the way to find out latent topic from documents or corpus available in multiple languages. When such heterogeneous data becomes huge, for example, Facebook reports about 6 billion new photo every month and 72 hours of video are uploaded to YouTube every minute, how to do unsupervised clustering or semi-supervised learning for such a huge heterogeneous data is becoming a challenging problem.

What is more, when the number of classes becomes higher and higher, there must be some correlation between classes. How to incorporate such a kind of correlation to boost the classification performance is attracting more and more attentions in nowadays machine learning research.

1.2 Notation

We summarize the notations and the definition of norms used in this paper. Matrices are written as uppercase letters and vectors are written as bold lowercase letters. For matrix $W = \{w_{ij}\}$, its i-th row, j-th column are denoted as $\mathbf{w}^i$, $\mathbf{w}_j$ respectively. The trace of the matrix W is denoted as $\mathrm{Tr}\,(W)$. The ℓ_p-norm of the vector $\mathbf{v} \in \mathbb{R}^n$ is defined as $\|\mathbf{v}\|_p = (\sum_{i=1}^{n} |v_i|^p)^{\frac{1}{p}}$, for $p \neq 0$ and the ℓ_0-norm of the vector $\mathbf{v}$, is defined as the number of non-zero entries of $\mathbf{v}$. The Frobenius norm of the matrix $W \in \mathbb{R}^{d\times m}$ is defined as $\|W\|_F = \sqrt{\sum_{i=1}^{d}\sum_{j=1}^{m} w_{ij}^2} = \sqrt{\sum_{i=1}^{d} \|\mathbf{w}^i\|_2^2}$. And the $\ell_{2,1}$-norm of matrix W is defined as $|W||_{2,1} = \sum_{i=1}^{d} \sqrt{\sum_{j=1}^{m} w_{ij}^2}$ and the $\ell_{2,0}$-norm of matrix W is defined as $||W||_{2,0} = \sum_{i=1}^{d} || \sum_{j=1}^{m} w_{ij}^2||_0$, where for a scalar a, $||a||_0 = 1$ if $a \neq 0$, $||a||_0 = 0$ if $||a|| = 0$. Please note that $\ell_{2,0}$-norm is not a valid norm because it does not satisfies the positive scalarbility: $||\alpha W||_{2,1} = |\alpha|||W||_{2,1}$ for any scalar α. The term "norm" here is for convenience.

This paper is organized as follows. Chaper II discusses several sparse learning models and how to use them to do feature selection on bio-data, where the number of features is much larger than the number of data point. Chapter III shows graph model to fuse multiple modality data to do unsupervised clustering or semi-supervised learning. When the number of data point is large, Chapter IV gives an efficient and robust multiple view K-Means clustering algorithm to release the burden of graph construction, clustering large-scale heterogeneous data. When the number of classes is large, Chapter V gives a global solution to low-rank linear regression and proves that the low-rank regression is equivalent to doing linear regression in LDA space. Chapter VI proposes the future work and summarize the thesis.

CHAPTER 2

SPARSE LEARNING BASED FEATURE SELECTION

2.1 Introduction

Generally speaking, feature ranking and feature selection algorithms may roughly be divided into three main types: filter, wrapper and embedded methods. These three basic categories differ in how the learning algorithm is incorporated in evaluating and selecting features. In filter methods, features are pre-selected by the intrinsic properties of the data without running the learning algorithm. Therefore, filter methods are independent of classifiers. Popular filter-type feature selection methods encompass F-statistic [6], reliefF [7], mRMR [3], t-test, Chi-square and information gain [8] and etc. [9] which all compute the sensitivity (correlation or relevance) of a feature with respect to (w.r.t.) the class label distribution of the data. These methods can be characterized by utilizing the global statistical information. In wrapper methods [10], feature selection is wrapped around predictors providing them subsets of features and receiving their feedback. Wrapper-type feature selection methods are tightly coupled with a specific classifier, such as correlation-based feature selection (CFS) [11], support vector machine recursive feature elimination (SVM-RFE) [12]. In spite of expensive computational cost, they often have good performance. In embedded methods, feature search and the learning algorithm are incorporated into a single optimization problem, which is also specific to the classifier. For example, Random multinomial logit (RMNL) [13].

With the development of sparsity regularization, dimension reduction has been widely investigated and applied into feature selection studies as well. For example, ℓ_1-norm SVM can perform variable selection via the ℓ_1-norm regularization [14], which tends to

give sparse solution to the following optimization problem. However, it has some limitations due to the fact that the number of selected features is upper bounded by the data sample size. What is more, since the sparsity nature of ℓ_1-norm does not discovery data's intrinsic group structure, it tends to pick up features without considering all the classes. In order to overcome the ℓ_1-norm's drawbacks, a method called Hybrid Huberized SVM (HHSVM) [15] was proposed combining both ℓ_1-norm and ℓ_2-norm regularization with the huberized hinge loss function to form a more flexible feature selection method. Nevertheless, it was designed only for binary case only. In multi-task learning, Obozinsky et al. [16] , Argyriou *et. al.* [17] have developed a $\ell_{2,1}$-norm based feature selection method that imposes the structure sparsity in feature selection, i.e. the selected features have large scores across all the tasks (classes) and the unselected features have small scores (sparse) over all tasks. However, due to the optimization difficulty in multi-class case, the approach used least square loss function instead of the hinge loss function.

In this chapter, we will propose three sparse learning feature selection methods to select features w.r.t multiple classes. In the following paragraph, we will introduce $\ell_{2,1}$-Norm Support Vector Machine first and then we will propose another practical feature selection approach called "Exact Top K Feature Selection Method with $\ell_{2,0}$-Norm Constraint". At last, if the number of classes is large, we propose another sparse learning method to

2.2 Multi-Class Feature Selection via $\ell_{2,1}$-Norm Support Vector Machine

2.2.1 Multi-Class Hinge Loss With $\ell_{2,1}$-Norm Regularization

As we know, hinge loss is usually better than the Least Square loss in terms of classification tasks [18]. In this section, we propose the following multi-class feature selection method based on hinge loss as well.

$$\min_{W} f(W^T X, Y) + \alpha \|W\|_{2,1} \tag{2.1}$$

where function f is the multi-class hinge loss function as defined as follows,

$$f(W^T X, Y) = \sum_{i=1}^{n} (1 - \mathbf{w}_{y_i}^T \mathbf{x}_i + \max_{m \neq y_i} \quad \mathbf{w}_m^T \mathbf{x}_i)_+ \tag{2.2}$$

and $\ell_{2,1}$-norm regularization term is defined as

$$\|W\|_{2,1} = \sum_{i=1}^{p} \sqrt{\sum_{j=1}^{k} w_{i,j}^2} = \sum_{i=1}^{p} \left\|\mathbf{w}^i\right\|_2 \tag{2.3}$$

Please note that as we defined in Chapter 1, $\mathbf{w}^i$ denotes the i-th row vector of matrix $W \in \mathbb{R}^{p \times k}$. Some other literatures called the $\ell_{2,1}$-norm as $\ell_{1,2}$-norm, or ℓ_2/ℓ_1-norm, or ℓ_1/ℓ_2-norm.

From the sparsity perspective, although the $\ell_{2,0}$-norm is more desirable, that is, $R(W) = \sum_{i=1}^{p} \|\mathbf{w}^i\|_2^0$, we will use $\ell_{2,1}$-norm based on the subsequent two reasons: On one hand, $\ell_{2,1}$-norm regularization term is convex and can be easily optimized [19]. On the other hand, it was shown that the results of ℓ_0-norm is identical or approximately identical to the ℓ_1-norm results under practical conditions [20]. So does $\ell_{2,0}$-norm and $\ell_{2,1}$-norm.

Here the key new development is the first time to combine multi-class hinge loss with $\ell_{2,1}$-norm regularization term to do the feature selection across all the classes, which has never been solved before due to its optimization difficulty. Although the hinge loss with $\ell_{2,1}-$norm regularization problem is a convex problem, complete solution path has not been provided yet due to the complexity of multi-class hinge loss as well as the non-smooth regularization term. In the next section, we will propose an efficient algorithm to tackle Eq. (2.1), with the proof of its convergence.

2.2.2 An Efficient Optimization Algorithm

Chih-Jen Lin et al. have solved the following"real" multi-class SVM problem with the published code [21].

$$\begin{aligned}
&\min_{\mathbf{w}_m,\xi_i} \tfrac{1}{2}\sum_{m=1}^{k}\mathbf{w}_m^T\mathbf{w}_m + \alpha\sum_{i=1}^{n}\xi_i \\
&s.t.\mathbf{w}_{y_i}^T\mathbf{x}_i - \mathbf{w}_m^T\mathbf{x}_i \geq e_{im} - \xi_i, \\
&\quad for\ i = 1, ..., n, m = 1, ..., k \\
&\quad \sum_{i=1}^{n}\xi_i \leq \alpha \\
&\quad \xi_i \geq 0,\ for\ i = 1, ..., n
\end{aligned} \tag{2.4}$$

In other words, given X and Y, we have a function to obtain W^*,

$$W^* = \arg\min_{W} f(W^T X, Y) + \alpha \left\|W\right\|_2^2 \tag{2.5}$$

where the function f is also defined in Eq. (2.2).

Let $J(W) = f(W^T X, Y) + \alpha\|W\|_{2,1}$. We find that the result of taking derivative of $J(W)$ w.r.t.W is equivalent to the derivative of the following objective function w.r.t W,

$$\min_{W} f(W^T X, Y) + \alpha \text{Tr}\left(W^T D W\right) \tag{2.6}$$

where D is the diagonal matrix of W, and the i-th element on the diagonal is defined as

$$d_{ii} = \frac{1}{2\|\mathbf{w}^i\|_2}, \forall\ i = 1, ..., p \tag{2.7}$$

Note that D is dependent to W. So it is also an unknown variable. We propose an iterative algorithm to find out the global solution W, that is, in each iteration, W is calculated with the current D and then D is updated according to the current W. The iteration procedure is repeated until the algorithm converges.

In order to do that, we need to change the variables, let $W_1 = D^{\frac{1}{2}}W$ and $X_1 = D^{-\frac{1}{2}}X$.

Therefore,

$$\begin{aligned}
&\min_{W} f(W^T X, Y) + \alpha \text{Tr}\,(W^T D W) \\
&= \min_{W} f(W^T D^{\frac{1}{2}} D^{-\frac{1}{2}} X, Y) + \alpha \text{Tr}\,(W^T D^{\frac{1}{2}} D^{\frac{1}{2}} W) \\
&= \min_{W_1} f(W_1^T X_1, Y) + \alpha \text{Tr}\,(W_1^T W_1)
\end{aligned} \tag{2.8}$$

Note that D is a diagnoal matrix. So far, we have bridged the new objective function Eq. (2.6) with the solvable objective function Eq. (2.5).

Algorithm 1 An efficient iterative algorithm to solve the optimization problem in Eq. (2.8)

Input: data $X \in \mathbb{R}^{p \times n}$, label $Y \in \mathbb{R}^{k \times n}$, regularization parameter α

Output: the coefficient matrix $W \in \mathbb{R}^{p \times k}$

Procedure:

1: Initialize the coefficient matrix

$W^{(0)} = \{w_{ij} = 1\}, \quad i = 1, ...p, \quad j = 1, ...k.$

2: Initialize the diagonal matrix $D^{(0)}$, where the i-th diagonal element is defined by Eq. (2.7).

3: Initialize matrix $W_1^{(0)}$ as $W_1^{(0)} = (D^{(0)})^{\frac{1}{2}} W^{(0)}$

4: Set $t = 0$

5: **repeat**

6: Relax the input data as $X_1^{(t)} = (D^{(t)})^{-\frac{1}{2}} X$

7: Calculate the coefficient matrix $W_1^{(t+1)} = \arg\min_{W} f((W_1^{(t)})^T X_1^{(t)}, Y) + \alpha \text{Tr}\,((W_1^{(t)})^T W_1^{(t)})$ by Crammer's Algorithm using LIBLINEAR [21].

8: Update the diagonal matrix $D^{(t+1)}$ by Eq. (2.7).

9: $t = t + 1$

10: **until** Converges

11: Calculate the output $W = {D^{(*)}}^{-\frac{1}{2}} W_1^{(*)}$

Please note that when $\mathbf{w}^i = 0$, then $d_{ii} = 0$ is a subgradient of $\|W\|_{2,1}$, w.r.t. $\mathbf{w}^i$. However we cannot set $d_{ii} = 0$ when $\mathbf{w}^i = 0$, otherwise the derived algorithm cannot be guaranteed to converge. To solve this issue, we can regularize d_{ii} as $d_{ii} = \frac{1}{2\sqrt{(\mathbf{w}^i)^T\mathbf{w}^i+\zeta}}$, where ζ is a very small number and in our experiment we will use eps in matlab as the value for ζ. The derived algorithm can be proved to minimize the regularized $\ell_{2,1}$-norm of W (defined as $\sum_{i=1}^{p}\sqrt{(\mathbf{w}^i)^T\mathbf{w}^i+\zeta}$) instead of $\ell_{2,1}$-norm of W. It is easy to verified thta the regularized $\ell_{2,1}$-norm of W approximates the $\ell_{2,1}$-norm of W when $\zeta \to 0$.

We summarize the proposed iterative method in Algorithm 6.

2.2.3 The Proof of The Convergence

We will utilize the following Theorem to prove the convergence of the Algorithm 1.

Theorem 1. *The Algorithm 6 will monotonically decrease the objective of the problem Eq. (2.8) in each iteration and converge to the global optimum of the problem.*

Proof. Since Crammer's Algorithm gives the solution to problem Eq. (2.5), we will find out the solution to the following problem by changing the variable:

$$\min_W f(W^TX, Y) + \alpha \text{Tr}\,(W^TDW) \tag{2.9}$$

where D is a function of W satisfying Eq. (5.26). Therefore, in the t-th iteration,

$$W^{(t+1)} = \arg\min_W f((W^{(t)})^TX, Y) + \alpha \text{Tr}\,((W^{(t)})^TD^{(t)}W^{(t)}) \tag{2.10}$$

which indicates

$$\begin{aligned} &f((W^{(t+1)})^TX, Y) + \alpha \text{Tr}\,((W^{(t+1)})^TD^{(t)}(W^{(t+1)})) \\ &\leq f((W^{(t)})^TX, Y) + \alpha \text{Tr}\,((W^{(t)})^TD^{(t)}(W^{(t)})) \end{aligned} \tag{2.11}$$

The above inequality can be extended as,

$$\begin{aligned} &f((W^{(t+1)})^TX, Y) + \alpha \sum_{i=1}^{p} \frac{\left\|(\mathbf{w}^{(t+1)})^i\right\|_2^2}{2\left\|(\mathbf{w}^{(t)})^i\right\|_2} \\ &\leq f((W^{(t)})^TX, Y) + \alpha \sum_{i=1}^{p} \frac{\left\|(\mathbf{w}^{(t)})^i\right\|_2^2}{2\left\|(\mathbf{w}^{(t)})^i\right\|_2} \end{aligned} \tag{2.12}$$

since $(\left\|\left(\mathbf{w}^{(t+1)}\right)^i\right\|_2 - \left\|\left(\mathbf{w}^{(t)}\right)^i\right\|_2)^2 \geq 0$, we can obtain the next inequality

$$\left\|\left(\mathbf{w}^{(t+1)}\right)^i\right\|_2 - \frac{\left\|\left(\mathbf{w}^{(t+1)}\right)^i\right\|_2^2}{2\left\|\left(\mathbf{w}^{(t)}\right)^i\right\|_2} \leq \left\|\left(\mathbf{w}^{(t)}\right)^i\right\|_2 - \frac{\left\|\left(\mathbf{w}^{(t)}\right)^i\right\|_2^2}{2\left\|\left(\mathbf{w}^{(t)}\right)^i\right\|_2} \tag{2.13}$$

So the following inequality holds:

$$\begin{aligned} &\alpha \sum_{i=1}^{p} (\left\|\left(\mathbf{w}^{(t+1)}\right)^i\right\|_2 - \frac{\left\|\left(\mathbf{w}^{(t+1)}\right)^i\right\|_2^2}{2\left\|\left(\mathbf{w}^{(t)}\right)^i\right\|_2}) \\ &\leq \alpha \sum_{i=1}^{p} (\left\|\left(\mathbf{w}^{(t)}\right)^i\right\|_2 - \frac{\left\|\left(\mathbf{w}^{(t)}\right)^i\right\|_2^2}{2\left\|\left(\mathbf{w}^{(t)}\right)^i\right\|_2}) \end{aligned} \tag{2.14}$$

Adding Eq. (2.12) and Eq. (2.14) together, we arrive at

$$\begin{aligned} &f((W^{(t+1)})^T X, Y) + \alpha \sum_{i=1}^{p} \left\|\left(\mathbf{w}^{(t+1)}\right)^i\right\|_2 \\ &\leq f((W^{(t)})^T X, Y) + \alpha \sum_{i=1}^{p} \left\|\left(\mathbf{w}^{(t)}\right)^i\right\|_2 \end{aligned} \tag{2.15}$$

By the definition of $\ell_{2,1}$-norm, we get

$$\begin{aligned} &f((W^{(t+1)})^T X, Y) + \alpha \left\|W^{(t+1)}\right\|_{2,1} \\ &\leq f((W^{(t)})^T X, Y) + \alpha \left\|W^{(t)}\right\|_{2,1} \end{aligned} \tag{2.16}$$

Thus the Algorithm 6 will monotonically decrease the objective of the problem in Eq. (2.1) in each iteration t. At last, it will converge and $W^{(t)}$ and $D^{(t)}$ will satisfy the Eq. (5.26) and Crammer's Algorithm. Furthermore, please note that the problem in Eq. (2.8) is a convex problem, which indicates that $W_1^{(*)}$ is a global optimum solution to the problem in Eq. (2.8) and W is the global optimum solution to the problem in Eq. (2.1). As a result, the Algorithm 6 will converge to the global optimum of the problem Eq. (2.1). □

Empirical results show that the convergence is fast and usually only a few iterations (less than 10) are needed to converge.

2.2.4 Experiments

2.2.4.1 Data Sets Description

To evaluate the performance of our $\ell_{2,1}$-norm SVM, we applied our Algorithm into five publicly available gene expression data sets and one Mass Spectrometry (MS) data set to do multi-class feature selection. All the data sets are standardized to be zero-mean and normalized by the standard deviation. The gene expression data sets are the malignant glioma (GLIOMA) data set [22], the human lung carcinomas (LUNG) data set [23], ALLAML data set [24], Human Carcinomas (Carcinomas) data set [25], mixed-lineage leukaemia (MLL) data set [26]. MS data is the Prostate Cancer data set. All of those data sets have the characteristic that the number of the samples is much less than the number of the features.

We give a brief description of all the data sets used in our subsequent experiments and summarize them in Table 2.1.

GLIOMA data set encompasses 50 samples of four classes in total: cancer glioblastomas (CG), non-cancer glioblastomas (NG), cancer oligodendrogliomas (CO) and non-cancer oligodendrogliomas (NO), which have 14, 14, 7, 15 samples, respectively. Each sample has 12625 genes. Genes with minimal variations across the samples were removed before the experiment. Also, intensity thresholds were set at 20 and 16,000 units for this data set. Genes whose expression levels varied less than 100 units between samples or varied less than 3 folds between any two samples were excluded. After preprocessing, we obtained a data with 50 samples and 4433 genes.

LUNG data contains 203 samples of five classes, which have 139, 21, 20, 6, 17 samples, respectively. Each sample has 12600 genes. In the preprocessing, the genes with standard deviations less than 50 expression units were removed and we got a data set with 203 samples and 3312 genes at last.

Table 2.1. Data set summary.

data name	# samples (n)	# features (p)	# classes (k)
GLIOMA	50	4433	4
LUNG	203	3312	5
ALLAML	72	7129	2
Carcinom	174	9182	11
MLL	72	12582	3
Pro-MS	89	15154	2

ALLAML data set contains 72 samples of two classes, that is, ALL and AML, which have 47 and 25 samples, respectively. Each sample contains 7,129 genes.

Carcinomas data set is composed of 174 samples of eleven classes, prostate, bladder/ureter, breast, colorectal, gastroesophagus, kidney, liver, ovary, pancreas, lung adenocarcinomas and lung squamous cell carcinoma, which have 26, 8, 26, 23, 12, 11, 7, 27, 6, 14, 14 samples, respectively. The raw data encompasses 12533 genes and the after preprocessing, the data set has 174 samples and 9182 genes.

MLL data set contains 72 samples of three classes, acute lymphoblastic leukaemia, acute myeloid leukaemia and mixed-lineage leukaemia, which have 24, 20 and 28 samples, respectively. Each sample has 12582 genes.

Prostate-MS data set consists of 89 samples of two classes, patient and normal people, which have 26 and 63 samples, respectively. Each mass spectrum is composed of peak amplitude measurements at 15154 points defined by a corresponding m/z value.

2.2.5 Experiment Setup

We compare our Algorithm ($\ell_{2,1}$-norm SVM) with six naive multi-class feature selection methods such as F-statistic [6] , reliefF [27], mRMR [3], t-test, Chi-square, information gain [8]. What is more, in order to demonstrate the power of the combination of

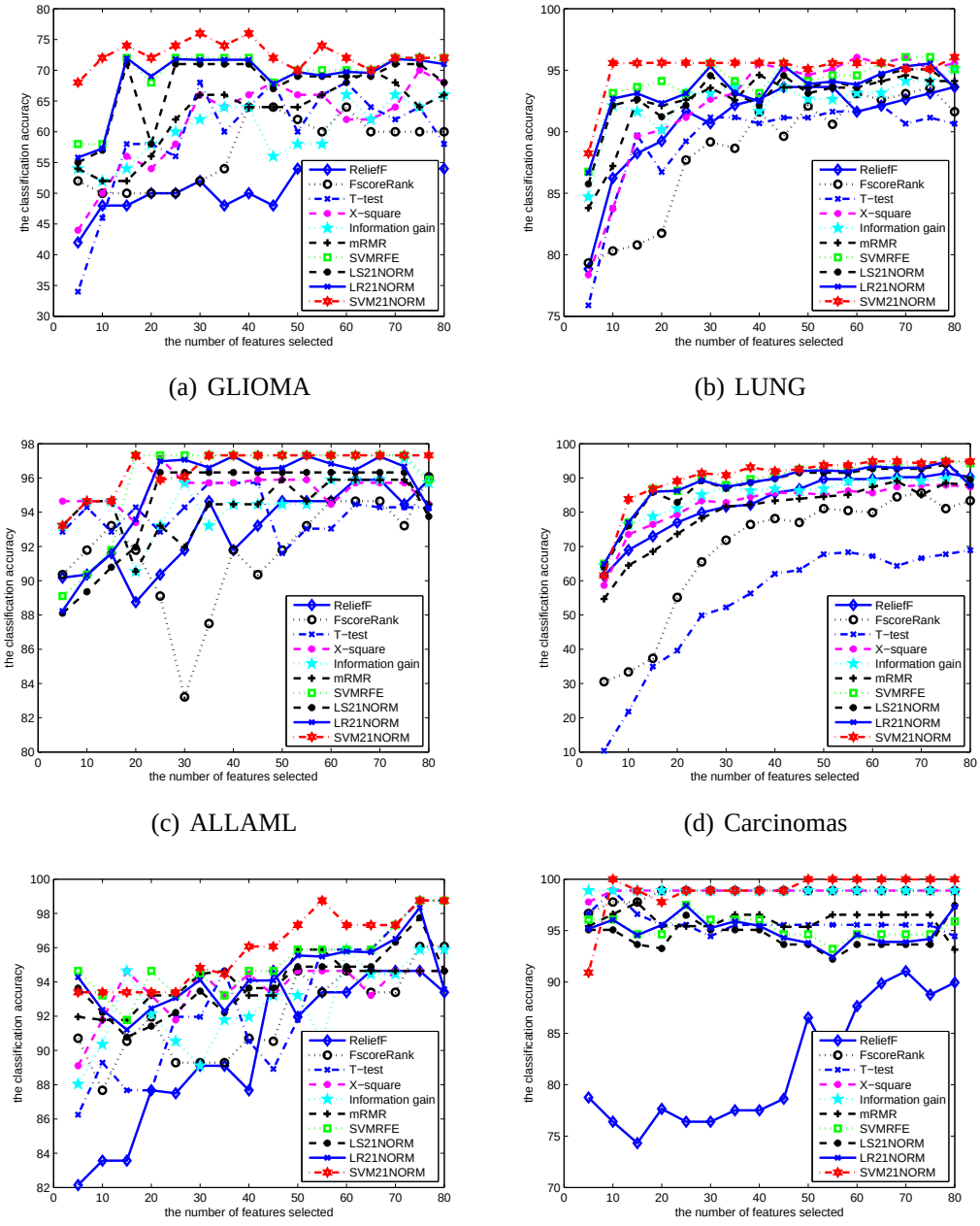
multi-class hinge loss with $\ell_{2,1}$ regularization term to do feature selection, we also compare three baseline methods. The famous multi-class SVMRFE [28] uses hinge loss with ℓ_2-norm regularization. $\ell_{2,1}$-norm LS method which uses Least Square loss with $\ell_{2,1}$-norm. $\ell_{2,1}$-norm LR method which uses logistic loss with $\ell_{2,1}$ regularization term.

Because we concern the multi-class feature selection method, we don't compare binary feature selection method, such as HHSVM [15]. Due to the upper bound and small number of samples data with 5-fold cross validation in our experiment, we do not consider ℓ_1-SVM neither. Regarding to multi-class SVMRFE, since our method resorts to Crammer and Singer's multi-class SVM (MSVMCS) with $\ell_{2,1}$-norm regulation, we will use MSVMCS to do the recursive feature elimination as well for fair comparison.

The Support Vector Machine (SVM) with linear kernel model, $C = 1$ and K nearest neighbor (KNN) with $K = 1$ will be used as two popular classifiers to evaluate the performances of different multi-class feature selection algorithms. In order to fixed number of selected features (from 5 to 80 with the incremental step size 5), we sort the row index of matrix W by the row summation value and features are selected by the top ranked indices. In addition, all the experiments are using 5-fold cross-validation and the average classification accuracy based on the above two classifiers are reported. As we know, when the penalty parameter α is large enough, it tends to reduce the coefficients of more irrelevant features to exactly zero [15]. Therefore, the larger α is, the more irrelevant features are eliminated from the model and we will get a more sparse matrix W. We utilized 2-fold cross validation inside the training data to decide the value of the regularization parameter empirically.

2.2.5.1 Classification Comparison Using Selected Features

Fig. 2.1 and Fig. 2.2 show the comparisons result of all nine multi-class feature selection methods in terms of classification accuracy on six data sets using SVM classifier

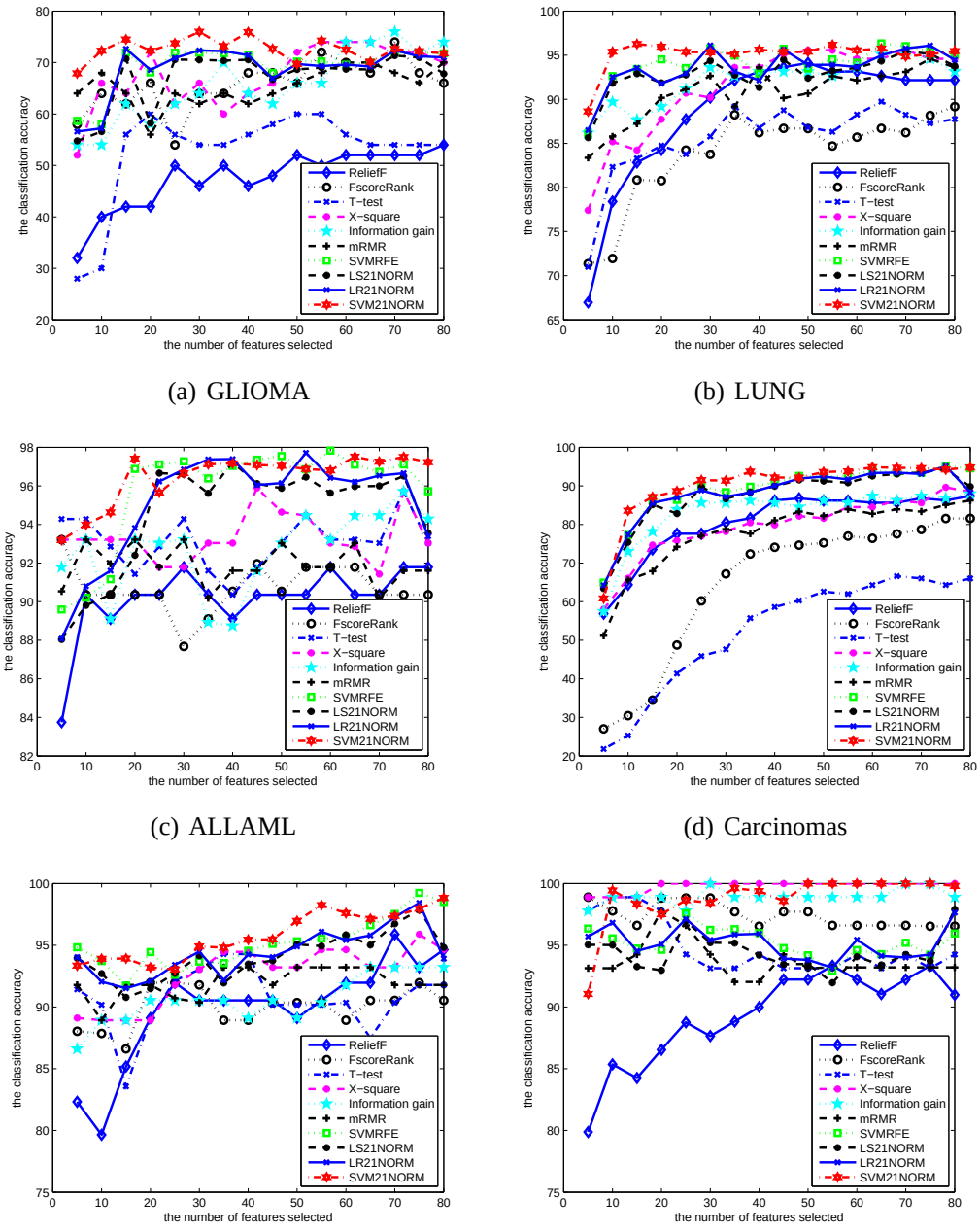


(a) GLIOMA (b) LUNG

(c) ALLAML (d) Carcinomas

(e) MLL (f) PROSTATE-MS

Figure 2.1. Comparisons of nine feature selection algorithms on six data sets in terms of classification accuracy using SVM as classifer with 5-fold cross-validation. SVM21NORM is our method whose curve are marked as red..



(a) GLIOMA (b) LUNG

(c) ALLAML (d) Carcinomas

(e) MLL (f) PROSTATE-MS

Figure 2.2. Comparisons of nine feature selection algorithms on six data sets in terms of classification accuracy using KNN ($K = 1$) as classifer with 5-fold cross-validation. SVM21NORM is our method whose curve are marked as red..

Table 2.2. Classification Accuracy of SVM using 5-fold cross validation evaluated on top 20 selected features. RF: ReliefF, F-s: F-score, CS: χ^2, IG: Information Gain, RFE: multi-class SVMRFE, LS21: Least Square loss with $\ell_{2,1}$-norm regularization, LR21: logistic loss with $\ell_{2,1}$-norm regularization and SVM21: $\ell_{2,1}$-norm SVM (our method).

	Average accuracy of top 20 features (%) SVM									
	RF	F-s	T-test	CS	IG	mRMR	RFE	LS21	LR21	SVM21
GLIOMA	50.00	50.00	58.00	54.00	58.00	56.00	54.00	58.00	70.00	**72.00**
LUNG	89.24	81.75	86.73	90.14	90.17	92.12	94.12	91.22	93.33	**95.61**
ALLAML	88.75	91.79	94.29	93.39	90.54	90.54	**97.32**	91.96	94.58	**97.32**
Carcinom	76.99	55.11	39.60	79.32	81.06	73.63	86.22	82.86	87.27	**89.09**
MLL	87.68	91.96	87.68	93.21	92.14	93.21	**94.64**	91.42	93.46	93.39
Pro-MS	77.65	**98.89**	95.56	**98.89**	**98.89**	95.42	94.64	93.26	96.54	97.77
Average	78.39	78.25	76.97	84.83	85.13	83.49	81.59	84.79	89.20	**90.87**

Table 2.3. Classification Accuracy of SVM using 5-fold cross validation evaluated on top 80 selected features. RF: ReliefF, F-s: F-score, CS: χ^2, IG: Information Gain, RFE: multi-class SVMRFE, LS21: Least Square loss with $\ell_{2,1}$-norm regularization, LR21: logistic loss with $\ell_{2,1}$-norm regularization and SVM21: $\ell_{2,1}$-norm SVM (our method).

	Average accuracy of top 80 features (%) SVM									
	RF	F-s	T-test	CS	IG	mRMR	RFE	LS21	LR21	SVM21
GLIOMA	60.00	58.00	68.00	66.00	66.00	**72.00**	**72.00**	68.00	**72.00**	**72.00**
LUNG	93.63	91.63	90.66	95.58	95.10	94.12	95.10	93.66	94.58	**96.07**
ALLAML	95.89	96.07	94.29	94.46	95.71	94.46	95.89	93.75	95.23	**97.32**
Carcinom	90.24	83.32	68.91	87.33	89.65	87.92	94.25	89.52	88.79	**94.82**
MLL	93.39	96.07	98.75	94.64	95.89	94.64	**98.75**	94.64	94.57	**98.75**
Pro-MS	89.93	98.89	94.44	98.89	98.89	93.14	95.89	97.44	98.33	**100**
Average	86.18	87.67	84.51	88.15	90.21	88.38	91.98	86.23	90.58	**93.16**

and KNN classifier respectively. Table 2.2 and Table 2.3 illustrate the detailed experimental results for top 20 and top 80 features for all feature selection approaches using SVM (linear kernel, $C = 1$) respectively. And Table 2.4 and Table 2.5 demonstrate the detailed experimental results for top 20 and top 80 features for all feature selection approaches using KNN ($K = 1$) respectively. From them, we can obviously see that our method outperforms the naive multi-class feature selection approaches and achieve competitive performance com-

Table 2.4. Classification Accuracy of KNN using 5-fold cross validation evaluated on top 20 selected features. RF: ReliefF, F-s: F-score, CS: χ^2, IG: Information Gain, RFE: multi-class SVMRFE, LS21: Least Square loss with $\ell_{2,1}$-norm regularization, LR21: logistic loss with $\ell_{2,1}$-norm regularization and SVM21: $\ell_{2,1}$-norm SVM (our method).

	Average accuracy of top 20 features (%) KNN									
	RF	F-s	T-test	CS	IG	mRMR	RFE	LS21	LR21	SVM21
GLIOMA	42.00	66.00	60.00	**72.00**	58.00	56.00	68.00	68.00	68.00	**72.00**
LUNG	84.27	80.78	84.75	87.73	89.17	90.17	94.53	91.86	91.77	**95.96**
ALLAML	90.36	90.36	91.43	93.21	93.21	93.21	96.88	92.40	93.82	**97.41**
Carcinom	77.61	48.76	41.34	75.85	83.92	74.22	86.37	82.83	86.92	**88.65**
MLL	89.11	91.96	89.11	88.93	90.54	91.79	94.45	91.48	92.10	**93.21**
Pro-MS	86.54	98.82	97.78	**100.00**	98.89	97.78	94.63	92.97	95.08	97.51
Average	78.31	79.45	77.40	86.29	85.62	83.86	89.15	84.96	88.04	**90.84**

Table 2.5. Classification Accuracy of KNN using 5-fold cross validation evaluated on top 80 selected features. RF: ReliefF, F-s: F-score, CS: χ^2, IG: Information Gain, RFE: multi-class SVMRFE, LS21: Least Square loss with $\ell_{2,1}$-norm regularization, LR21: logistic loss with $\ell_{2,1}$-norm regularization and SVM21: $\ell_{2,1}$-norm SVM (our method).

	Average accuracy of top 80 features (%) KNN									
	RF	F-s	T-test	CS	IG	mRMR	RFE	LS21	LR21	SVM21
GLIOMA	54.00	66.00	54.00	62.00	58.00	62.00	**72.00**	70.00	**72.00**	**72.00**
LUNG	92.17	89.17	84.75	92.58	95.10	94.12	95.07	93.78	94.46	**95.47**
ALLAML	91.79	90.36	91.43	94.46	95.71	94.46	95.72	93.55	93.39	**97.23**
Carcinom	87.36	81.58	41.34	85.31	86.33	85.43	94.55	89.80	88.31	**94.78**
MLL	94.64	90.54	90.54	92.64	95.89	92.64	98.51	94.82	93.92	**98.83**
Pro-MS	90.98	96.54	96.54	98.89	98.89	95.14	95.95	97.90	97.64	**99.80**
Average	85.16	85.70	81.14	90.12	90.25	87.74	91.93	89.61	89.79	**92.98**

pared with multi-class SVMRFE especially if we only selection the top 20 features. Also, compared with multi-class Least Square loss or multi-class logistic loss, multi-class hinge loss with $\ell_{2,1}$-norm usually achieves the best performance.

Fig. 2.3 demonstrates the learned matrix (the solution to W of Eq. (2.6)) of data GLIOMA. Columns represent 4 classes and rows represent around 3000 features. The

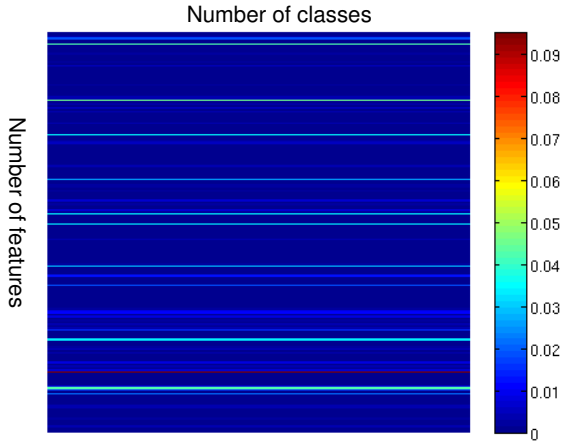


Figure 2.3. The learned matrix W of data GLIOMA.

brighter pixel means higher value entry. From it we can observe the intrinsic structural horizontal pattern of $\ell_{2,1}$-norm clearly, that is, selecting features for all the classes.

2.2.5.2 Algorithm Time Complexity Analysis

In Alg.6, LIBLINEAR [21] provides the solver for Eq. (2.5) using very efficient coordinate descent method and usually we can obtain the result for step 7 in less than 1 second for large scale data (like our bio-data). And since D is a diagonal matrix, the computational complexity in step 6 is also low. We report the average number of iteration and the average feature selection time for multi-class SVMRFE and our method using 5-fold cross validation in Table 2.6 for all the data used in our experiment, where we used Matlab $2009b$ and the configuration of our PC is Intel Corel 2 Duo CPU $E7300$ $2.66GHz$. Choosing 0.001 as the stop criterion, we can see that the converge rate of our method is fast and although our method can achieve competitive classification performance compared with multi-class SVMRFE, the speed of our method is much faster.

Table 2.6. Time comparison of our method and multi-class SVMRFE

data	iter# our	iter# RFE	T our (sec)	T RFE (sec)
G	12	4433	14.91	1115.73
L	30	3312	51.92	1023.86
A	13	7129	34.83	2127.95
C	15	9182	91.42	3192.84
M	13	12582	77.64	4244.23
P	30	15154	239.74	5256.38

G:GLIOMA, L: LUNG, A: ALLAML, C: Carcinom, M: MLL, P: Pro-MS

2.3 Exact Top-K Multi-Class Feature Selection via $\ell_{2,0}$-Norm Constraint

Since we are focusing on multi-class feature selection, structural sparsity regularization is desired, which can select the features across all the classes with jointly sparsity, *i.e.* each feature has either small score or large score for all the classes. From the sparsity perspective, although $\ell_{2,0}$-norm is more desirable, due to its nonconvex and non-smooth properties which will induce great difficulty in optimization, people prefer the convex $\ell_{2,1}$-norm as the regularization term [29] [19] [30]. As we know, such kind of approximation is under the assumption that the effects of $\ell_{2,0}$-norm regularization is identical or approximately identical to the $\ell_{2,1}$-norm. Nevertheless, the above assumption does not always hold in the real application [31]. Moreover, since the regularization parameter of $\ell_{2,1}$-norm does not have explicit meaning, for different data, it may change dramatically and people need to carefully tune its value based on the training data, which will take long time. Lots of related work of sparse learning based feature selection methods adopt the model based on convex problem due to the fact that convex problem has global solution. However, is it always true that the method based on convex problem is always better than that based on non-convex problem?

In this section, we will propose an efficient, robust and pragmatic multi-class feature selection model, which has the following advantages: (1) We show that it is NOT true that the feature selection method based on convex problem is always better than its counterpart

based on non-convex problem. (2) We tackle the original sparse problem with $\ell_{2,0}$-norm constraint directly instead of its relaxation or approximation problem. Therefore, we can get a more accurate solution. (3) Since there is only one term in the objective function, we avoid the computational burden of tuning the parameter for regularization term, which is desired for solving the real problem. (4) We are the first to provide an efficient algorithm to tackle the minimization problem of $\ell_{2,1}$-norm loss with the $\ell_{2,0}$-norm constraint. Extensive experiments on four benchmark biological datasets show that our approach outperforms the relaxed or approximate counterparts and state-of-art feature selection methods evaluated in terms of classification accuracy using two popular classifiers.

2.3.1 Sparse Learning Based Feature Selection Background

Typically, many sparse based supervised binary feature selection methods that arise in data mining and machine learning can be written as the approximation or relaxed version of the following problem:

$$\begin{aligned} &< \mathbf{w}^*, b >= \min_{\mathbf{w},b} ||\mathbf{y} - X^T\mathbf{w} - b\mathbf{1}||_2^2 \\ &s.t. ||\mathbf{w}||_0 = k \end{aligned} \tag{2.17}$$

where $\mathbf{y} \in \mathbb{B}^{n\times 1}$ is the binary label, $X \in \mathbb{R}^{d\times n}$ is the training data, $\mathbf{w} \in \mathbb{R}^{d\times 1}$ is the learned model, b is the learned biased scalar, $\mathbf{1} \in \mathbb{R}^{n\times 1}$ is a column vector with all 1 entries, and k is the number of the feature selected. Solving Eq. (2.17) directly has been approved NP-hard, very difficult in optimization. In many practical situations it is convenient to allow for a certain degree of error, and we can relax the optimization constraint using the following formulation,

$$< \mathbf{w}^*, b >= \arg\min_{\mathbf{w},b} \{||\mathbf{w}||_0 + \lambda||\mathbf{y} - X^T\mathbf{w} - b\mathbf{1}||_2^2\} \tag{2.18}$$

which is equivalent to the following "fidelity loss plus regularization" format,

$$< \mathbf{w}^*, b >= \arg\min_{\mathbf{w},b} \{||\mathbf{y} - X^T\mathbf{w} - b\mathbf{1}||_2^2 + \lambda||\mathbf{w}||_0\} \tag{2.19}$$

where $\lambda \in \mathbb{R}^+$ is the regularization parameter. Unfortunately, the way to tackle Eq. (2.19) is still challenging. To overcome this problem, the subsequent alternative formulation using ℓ_1-norm regularization instead of ℓ_0-norm has been proposed,

$$< \mathbf{w}^*, b >= \arg\min_{\mathbf{w},b} \{||\mathbf{y} - X^T\mathbf{w} - b\mathbf{1}||_2^2 + \lambda||\mathbf{w}||_1\} \tag{2.20}$$

After we get $\mathbf{w}^*$, we choose the feature indices corresponding to top k largest values of the summation of absolute values along each row. In statistic, people call Eq. (2.20) as the regularized counterpart of LASSO problem, which has been widely studied and proved to have a closed form solution.

Although people can use heuristic strategy, i.e. one V.S. all or one V.S. one to extend the above binary sparse based feature selection method to do multi-class feature selection, some structural sparsity is preferred, if the goal is to select features across all the classes. In multi-task learning, Obozinsky *et al.* and Argyriou *et. al.* [17] [32] have developed a $\ell_{2,1}$-norm square regularization term to couple feature selection across tasks.

2.3.2 Robust And pragmatic Multi-class Feature Selection

Given training data $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\} \in \mathbb{R}^{d\times 1}$ and its corresponding class labels $\{y_1, y_2, \cdots, y_n\} \in \mathbb{R}^{m\times 1}$, traditional least square regression solves the following optimization problem to learn the projection matrix $W \in \mathbb{R}^{d\times m}$ and the bias $\mathbf{b} \in \mathbb{R}^{m\times 1}$:

$$< W^*, \mathbf{b} >= \arg\min_{W,\mathbf{b}} \sum_{i=1}^{n} ||\mathbf{y}_i - W^T\mathbf{x}_i - \mathbf{b}||_2^2. \tag{2.21}$$

Since there is inevitable noise existing in the training data, in order to be robust to outliers, our proposed method will use the robust loss function:

$$< W^*, \mathbf{b} >= \arg\min_{W,\mathbf{b}} \sum_{i=1}^{n} ||\mathbf{y}_i - W^T\mathbf{x}_i - \mathbf{b}||_2, \tag{2.22}$$

which has a rotational invariant property whereas the pure ℓ_1-norm loss function does not has such desirable property [33]. In addition, for the sake of obtaining a more accurate model, we use $\ell_{2,0}$-norm constraint instead of impose it as the regularization term.

Denoting n training data $X \in \mathbb{R}^{d\times n}$ as well as the associated class labels $Y \in \mathbb{R}^{n\times m}$ for m classes, in this paper, we propose the following objective function to select k features in multi-class problems

$$\begin{aligned} &\min_{W,\mathbf{b}} |||Y - X^T W - \mathbf{1}\mathbf{b}^T||_{2,1} \\ &s.t. ||W||_{2,0} = k, \end{aligned} \tag{2.23}$$

where, $\mathbf{1} \in \mathbb{R}^{n\times 1}$ is a column vector with all its entries being 1.

2.3.3 Optimization Algorithm

In this section, we will propose an efficient algorithm to tackle Eq. (4.25) directly followed by the proof of its convergence to local solution.

2.3.3.1 General Augmented Lagrangian Multiplier Method

In [34], the general method of augmented Lagrange multipliers is introduced for solving constrained optimization problems of the kind:

$$\min_X f(X), \qquad s.t. \quad \text{Tr}\,(h(X)) = 0, \tag{2.24}$$

One may define the augmented lagrangian function:

$$L(X, \Lambda, \mu) = f(X) + \text{Tr}\,(\Lambda^T h(X)) + \frac{\mu}{2}||h(X)||_F^2, \tag{2.25}$$

where matrix Λ is the Lagrange multiplier and μ is a positive scalar called the quadratic penalty parameter and then Eq. (2.25) can be solved via the method of augmented Lagrange multipliers, outlined as Alg. 2.

2.3.3.2 Problem Reformulation

According to Augmented Lagrangian Multiplier (ALM) Method, we introduce two slack variables *i.e.* V and E. Eq. (4.25) can be reformulated as

$$\begin{aligned}\min_{W,\mathbf{b},V,\|V\|_{2,0}=k,E} &\|E\|_{2,1} + \tfrac{\mu}{2}\left\|W - V + \tfrac{1}{\mu}\Lambda\right\|_F^2 \\ &+\tfrac{\mu}{2}\left\|X^TW + \mathbf{1}\mathbf{b}^T - Y - E + \tfrac{1}{\mu}\Sigma\right\|_F^2\end{aligned} \tag{2.26}$$

2.3.3.3 An Efficient Algorithm to Solve the Constrained Problem

We will introduce an efficient algorithm based on the general ALM to tackle problem Eq. (2.26) alternatively and iteratively.

The first step is fixing W, V and E, solving $\mathbf{b}$. Then we need to solve the following subproblem:

$$\frac{\mu}{2}\left\|X^TW + \mathbf{1}\mathbf{b}^T - Y - E + \frac{1}{\mu}\Sigma\right\|_F^2 \tag{2.27}$$

Take derivative w.r.t. $\mathbf{b}$ and set it to zero, we have

$$\mathbf{b} = \frac{1}{n}(Y + E - \frac{1}{\mu}\Sigma)^T\mathbf{1} - \frac{1}{n}W^TX\mathbf{1} \tag{2.28}$$

The second step is fixing V, $\mathbf{b}$ and E, solving W. Then the objective function becomes,

$$\min_W \left\|W - V + \frac{1}{\mu}\Lambda\right\|_F^2 + \left\|X^TW + \mathbf{1}\mathbf{b}^T - (Y + E - \frac{1}{\mu}\Sigma)\right\|_F^2 \tag{2.29}$$

Take derivative w.r.t. W and set it to zero, we have

$$W = (XX^T + I)^{-1}(V - \frac{1}{\mu}\Lambda + X(Y + E - \frac{1}{\mu}\Sigma - \mathbf{1}\mathbf{b}^T)) \tag{2.30}$$

where $I \in \mathbb{R}^{d\times d}$ is the identity matrix.

The third step is fixing W, $\mathbf{b}$ and E, solving V. The subproblem becomes,

$$\min_{\|V\|_{2,0}=k} \left\|V - (W + \frac{1}{\mu}\Lambda)\right\|_F^2 \tag{2.31}$$

which can be solved by Alg. 3.

The fourth step is fixing W, $\mathbf{b}$ and V, solving E. The subproblem becomes,

$$\min_E \frac{1}{2}\left\|E-(X^TW+\mathbf{1}\mathbf{b}^T-Y+\frac{1}{\mu}\Sigma)\right\|_F^2+\frac{1}{\mu}\|E\|_{2,1} \tag{2.32}$$

Denote

$$G=X^TW+\mathbf{1}\mathbf{b}^T-Y+\frac{1}{\mu}\Sigma. \tag{2.33}$$

Then Eq. (2.32) is equivalent to the following problem,

$$\min_E \frac{1}{2}||E-G||_F^2+\frac{1}{\mu}||E||_{2,1}, \tag{2.34}$$

which can be decoupled as,

$$\min_{\mathbf{e}^i} \sum_{i=1}^{n} \frac{1}{2}||\mathbf{e}^i-\mathbf{g}^i||_2^2+\frac{1}{\mu}||\mathbf{e}^i||_2 \tag{2.35}$$

where $\mathbf{e}^i$ and $\mathbf{g}^i$ is the i-th row of matrix E and G respectively. And the solution to Eq. (2.35) is

$$\mathbf{e}^i=\begin{cases}(1-\frac{1/\mu}{||\mathbf{g}^i||_2})\mathbf{g}^i, & ||\mathbf{g}^i||_2>1/\mu\\ \mathbf{0}, & ||\mathbf{g}^i||_2\leq 1/\mu\end{cases} \tag{2.36}$$

We iteratively and alternatively update $\mathbf{b}, W, V, E$ according to the above four steps and summarize the whole Algorithm in Alg. 4.

2.3.3.4 Algorithm Analysis

Since Eq. (2.26) is not a convex problem, in each iteration, given fixed Λ, Σ, and μ, Alg. 4 will find its local solution. The convergence of ALM algorithm was proved and discussed in previous papers. Please refer to the literature therein [35] [36].

The overall computation complexity of our method is low, although we solve it separately and iteratively. In each iteration, the only computation burden is in Eq. (2.30), where we need to calculate an inverse $d \times d$ matrix. However, since it is only related to the input data, we can calculate it before we go to the loop. What is more, when the number

Algorithm 2 General Method of Augmented Lagrange Multiplier

Initialization:

1. Set $t = 0$

2. Initialize the Lagrangian multiplier matrix $\Lambda^{(t)}$.

3. Initialize the quadratic penalty parameter $\mu^{(t)}$.

4. Initialize the incremental step size parameter $\rho \geq 1$.

repeat

1. Update $X^{(t+1)} = \arg\min_{X} L(X^{(t)}, \Lambda^{(t)}, \mu^{(t)})$

2. Update $\Lambda^{(t+1)} = \Lambda^{(t)} + \mu^{(t)} h(X^{(t+1)})$

3. Update $\mu^{(t+1)} = \rho \mu^{(t)}$

4. Update $t = t + 1$

until Converges

Output: X^*

of feature is much larger than the number of data, we can resort to Woodbury formula to transform it as a $n \times n$ inverse matrix. Although its solution depends on the initialization, in the following experiment section, we will conduct experiment to demonstrate that its local solution is stable and its feature selection performance is better than that of some state-of-art sparse feature selection methods based on convex problems.

2.3.4 Experiment

We denote our proposed method as $\ell_{2,0}$-norm ALM. The performance of $\ell_{2,0}$-norm ALM is evaluated on four biological gene expression datasets. We give a brief description of all the datasets used in our subsequent experiments.

2.3.4.1 Datasets Descriptions

The gene expression datasets are the leukimia (LEU) data set [22], the human lung carcinomas (lUNG) data set [23], ALLA data set [24] and Human Carcinomas (Carcino-

Algorithm 3 The algorithm to solve Eq. (2.31)

Input:

1. The projection matrix W.
2. The Lagrangian multiplier matrix Λ
3. The quadratic penalty parameter μ.
4. The number of feature selected k.

Process:

1. Calculate $\widetilde{W} = W + \frac{1}{\mu}\Lambda$.
2. Calculate the vector $\mathbf{p} \in \mathbb{R}^{d\times 1}$, where each entry defined as $p_i = \sum_j \widetilde{w_{ij}}^2, \forall i = 1, 2, \cdots, d$.
3. Sort $\mathbf{p}$, find out the indices vector $\mathbf{q} = [q_1, q_2, \cdots, q_k]^T$ corresponding to top k sorted entries.
4. Assign i-th row of $\widetilde{W}$ to V if $i \in \mathbf{q}$;

assign zero row vector $\mathbf{0}^T \in \mathbb{R}^{1\times m}$to V, if $i \notin \mathbf{q}$.

Output: The slack variable matrix V.

mas) data set [25]. All these four datasets are standardized to zero-mean and normalized by the standard deviation, which are summarized in Table 2.7.

LEU data set encompasses two classes samples: 25 leukimia patient (Positive), 47 healthy patient (Negative). Each sample has 3571 genes. Genes with minimal variations across the samples were removed before the experiment. Also, intensity thresholds were set at 20 and $16,000$ units for this data set. After preprocessing, we obtained a data with 72 samples and 3571 genes.

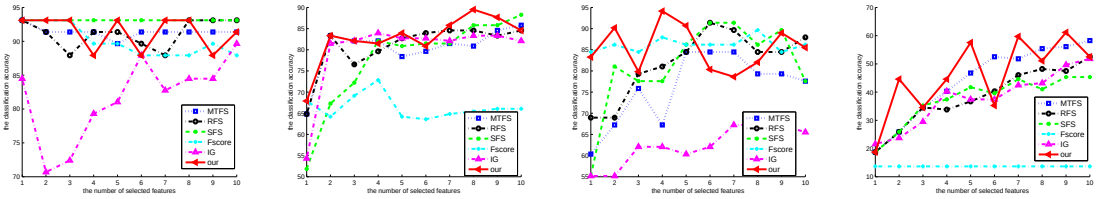
LUNG data contains 203 samples of five classes, which have 139, 21, 20, 6, 17 samples, respectively. Each sample has 12600 genes. In the preprocessing, the genes with standard deviations less than 50 expression units were removed and we got a data set with 203 samples and 3312 genes at last.

ALLA data set contains 72 samples of two classes, that is, ALL and AML, which have 47 and 25 samples, respectively. Each sample contains $7,129$ genes.

Carcinomas (CAR)data set is composed of 174 samples of eleven classes, prostate, blad-

Table 2.7. Gene Expression data set summary.

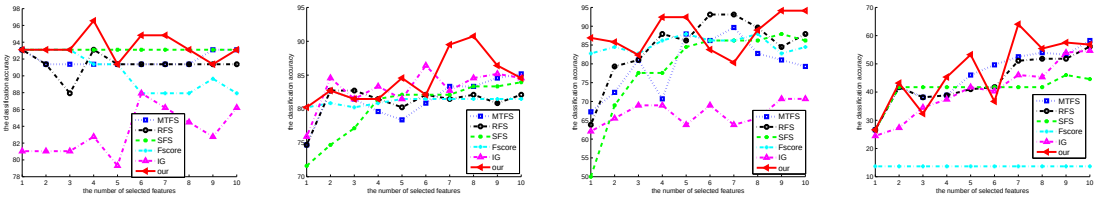
data name	# samples	# features	# classes
LEU	72	3571	2
LUNG	203	3312	5
ALLAML	72	7129	2
CAR	174	9182	11



(a) LEU (b) LUNG (c) ALLA (d) CAR

Figure 2.4. The classification accuracy using selected features by KNN.

der/ureter, breast, colorectal, gastroesophagus, kidney, liver, ovary, pancreas, lung adenocarcinomas and lung squamous cell carcinoma, which have 26, 8, 26, 23, 12, 11, 7, 27, 6, 14, 14 samples, respectively. The raw data encompasses 12533 genes and the after preprocessing, the data set has 174 samples and 9182 genes.



(a) LEU (b) LUNG (c) ALLA (d) CAR

Figure 2.5. The classification accuracy using selected features by SVM.

2.3.4.2 Experiment Setup

In our experiments, for each data, we will randomly select 20% to do the training and use the remaining part as testing. The reason we use smaller portion of training data is because it is well known that when the number of training data becomes sufficiently large, any feature selection method will work well. We select the number of features ranging from 1 to 10 with the incremental step 1 and the feature selection performance is evaluated by average classification accuracy on two popular classifiers, *i.e.* K nearest neighbor (KNN) and support vector machine (SVM). Specifically, we set up KNN with $K = 1$ and SVM with linear kernel $C = 1$ respectively for their intuitive meaning and simplicity. Here we assume that the better the feature selection algorithm is, the higher classification accuracy we will get. We compare our feature selection method with the following two basic filter methods:

Fisher Score [37] selects each feature independently according to the score under the Fisher criterion.

Information Gain (IG) [8] computes the sensitivity (correlation or relevance) of a feature w.r.t the class label distribution of the data.

In addition, we also compare our approach with some similar feature selection methods based on sparse learning:

Multi-Task Feature Selection (MTFS) [38] selects features across multi-task (multi-class) by solving a general loss function with $\ell_{2,1}$-norm regularization. .

Robust Feature Selection (RFS) [19] selects features w.r.t multi-class and can be robust to the outlier data by solving a joint $\ell_{2,1}$-norm problem.

Sparse Feature Selection (SFS) [39] selects features by solving a smoothed general loss function with a more sparse $\ell_{2,0}$-norm constraint.

We tune the regularization parameter in MTFS and RFS to let the non-zero row number of

Table 2.8. The mean and std of the converged objective function value of our method using 50 random initialization

data	$k = 1$	$k = 5$	$k = 8$	$k = 10$
LEU	1.72 ± 1.80	0.68 ± 0.49	0.58 ± 0.43	0.66 ± 0.37
LUNG	23.61 ± 2.87	11.81 ± 0.38	11.31 ± 0.84	10.12 ± 0.74
ALLA	6.06 ± 2.26	2.45 ± 1.24	1.28 ± 0.85	0.92 ± 0.56
CAR	29.69 ± 1.73	23.01 ± 1.70	19.81 ± 1.68	18.40 ± 1.48

the optimum solution W exactly equal to the number of selected features. Because MTFS and RFS both solve a convex optimization problem, they will get global solution finally. However, SFS and our method are based on $\ell_{2,0}$-norm constraint and we can only find local solution. In our experiment, we used the optimum solution of MTFS as the initialization for SFS and used random initialization for our method. Since there is an explicit meaning of the constraint k in our method or SFS, we can avoid the heavy burden of tuning regularization parameter and just make them as the number of selected features. We use the following parameters $\mu = 0.01$, $\rho = 1.02$ and choose 1000 as the maximum number of iterations in Alg. 4.

2.3.4.3 Feature Selection Results

Fig. 2.4 shows the classification accuracy V.S. the number of selected feature using KNN classifier. Similarly, Fig. 2.5 demonstrates the feature selection results by SVM. From them, we can see that when the number of selected feature is small, particularly the one with less than 5 features, the classification result of our method can beat MTFL as well as RFS consistently, since our method can find a more sparse solution by $\ell_{2,0}$-norm constraint instead of the solution to the relaxed regularization problem. Because SFS finds local solution, its performance depends on the initialization, *i.e.* MTFS. When feature selection result of MTFS is good, like LEU data, SFS can achieve very promising results. However, for some data, like LUNG, when MTFS performs badly, SFS will stuck at the bad local optimum. When the number of selected feature increases, all the sparse learning based

feature selection methods will tend to perform similarly, which is within our expectation. Next we will conduct experiment to show that our method can find stable local solutions under different random initializations.

2.4 Conclusion

In classification problem, the large number of features and the relatively small number of data samples pose great challenges for classification. To tackle these problems, in this chapter, we proposed a novel and efficient multi-class feature selection method with emphasizing the combination of multi-class hinge loss and $\ell_{2,1}$-norm regularization minimization ($\ell_{2,1}$-norm SVM) or least square loss with $\ell_{2,0}$-norm constraint. The $\ell_{2,1}$-norm or $\ell_{2,0}$-norm can capture the joint sparse structure to select features across all the classes, which naturally solves the feature selection for multi-class problem. An efficient algorithm with proved convergence has been provided and broad empirical studies have been performed on the bench mark data sets. Compared with some of the existing the state-of-art methods, our method can consistently achieves better multi-class feature selection performance evaluated on two popular classifiers.

Algorithm 4 The algorithm to solve Eq. (2.26)

Input:

1. Training data $X_{tr} \in \mathbb{R}^{d \times n_{tr}}$, training labels $Y_{tr} \in \mathbb{R}^{n_{tr} \times m}$

2. The number of feature selected k.

3. The initial projection matrix W_0.

Output:

1. The k selected feature indices vector **q**.

2. The objective function value obj

3. The learned projection matrix W and bias **b**.

Initialization:

1. Set $t = 0$

2. Initialize the projection matrix as $W = W_0$.

3. Initialize the Lagrangian multiplier matrix $\Lambda \in 0^{d \times m}$, $\Sigma \in 0^{n \times m}$.

4. Initialize the quadratic penalty parameter $\mu = 0.1$.

5. Initialize the incremental step size parameter $\rho = 1.02$.

Process:

repeat

1. Update the bias **b** by Eq. (2.28).

2. Update the projection matrix W by Eq. (2.30).

3. Update the the slack variable matrix V by Alg. 3.

4. Calculate G by Eq.(2.33).

5. Update E by Eq. (2.36).

6. Update $\Lambda^{(t+1)} = \Lambda^{(t)} + \mu^{(t)}(W^{(t+1)} - V^{(t+1)})$

7. Update $\Sigma^{(t+1)} = \Sigma^{(t)} + \mu^{(t)}(X^T W^{(t+1)} + \mathbf{1}\mathbf{b}^{(t+1)^T} - Y - E^{(t+1)})$

8. Update $\mu^{(t+1)} = \rho\mu^{(t)}$

9. Update $t = t + 1$

until Converges

CHAPTER 3

MULTI-VIEW K-MEANS CLUSTERING ON BIG DATA

3.1 Introduction

With the rising of data sharing websites, such as Facebook and Flickr, there is a dramatic growth in the number of data. For example, Facebook reports about 6 billion new photo every month and 72 hours of video are uploaded to YouTube every minute. One of major data mining tasks is to unsupervised categorize the large-scale data [40–43], which is useful for many information retrieval and classification applications. There are two main computational challenges in large-scale data clustering: (1) How to integrate the heterogeneous data features to improve the performance of data categorizations? (2) How to reduce the computational cost of clustering algorithm for large-scale applications?

Many scientific data have heterogeneous features, which are generated from different data collection sources or feature construction ways. For example, in biological data, each human gene can be measured by different techniques, such as gene expression, Single-nucleotide polymorphism (SNP), Array-comparative genomic hybridization (aCGH), methylation; in visual data, each image/video can be represented by different visual descriptors, such as SIFT [44], HOG [45], LBP [46], GIST [47], CENTRIST [48], CTM [49]. Each type of features can capture the specific information in the data. For example, in visual descriptors, CTM uses the color spectral information and hence is good for categorizing the images with large color variations; GIST achieves high accuracy in recognizing natural scene images; CENTRIST is good for classifying indoor environment images; HOG can describe the shape information of the image; SIFT is robust to image rotation, noise, illumination changes; and LBP is a powerful texture feature. It is crucial to

integrate these heterogeneous features to create more accurate and more robust clustering results than using each individual type of features.

Although several graph based multi-view clustering algorithms were presented with good performance, they have the following two main drawbacks. On one hand, because all of them are graph based clustering method, the construction of data graph is a key issue. Using different kernels to build the graph will affect the final clustering performance a lot. Moreover, for some specific kernels, we have to consider the impact of the choice of parameters, such that the clustering results are sensitive to the parameters tuning. On the other hand, more important, due to the heavy computation of the kernel construction as well as eigen decomposition, these graph based methods cannot be utilized to tackle large-scale data clustering problem.

The classical K-means clustering is a centroid-based clustering method, which partitions the data space into a structure known as Voronoi diagram. Due to its low computational cost and easily parallelized process, the K-means clustering method has often been applied to solve large-scale data clustering problems, instead of the spectral clustering. However, the K-means clustering was designed for solving single-view data clustering problem. In this section, we propose a new robust multi-view K-means clustering method to integrate heterogeneous features for clustering. Compared to related clustering methods, our proposed method consistently achieves better clustering performances on six benchmark data sets. Our contributions in this paper are summarized in the following four folds:

(1) We propose a novel robust large-scale multi-view K-means clustering approach, which can be easily parallelized and performed on multi-core processors for big visual data clustering;

(2) Using the structured sparsity-inducing norm, $\ell_{2,1}$-norm, the proposed method is robust to data outliers and can achieve more stable clustering results with different initializations;

(3) We derive an efficient algorithm to tackle the optimization difficulty introduced by the non-smooth norm based loss function with proved convergence;

(4) Unlike the graph based algorithms, the computational complexity of our methods is similar to the standard K-means clustering algorithm. Because our method does not require the graph construction as well as the eigen-decomposition, it avoids the heavy computational burden and can be used for solving large-scale multi-view clustering problems.

3.2 Robust Multi-View K-Means Clustering

As one of most efficient clustering algorithms, K-means clustering algorithm has been widely applied to large-scale data clustering. Thus, to cluster the large-scale multi-view data, we propose a new robust multi-view K-means clustering (RMKMC) method.

3.2.1 Clustering Indicator Based Reformulation

Previous work showed that the G-orthogonal non-negative matrix factorization (NMF) is equivalent to relaxed K-means clustering [50]. Thus, we reformulate the K-means clustering objective using the clustering indicators as:

$$\begin{aligned} &\min_{F,G} ||X^T - GF^T||_F^2 \\ &s.t.\ G_{ik} \in \{0,1\},\ \sum_{k=1}^{K} G_{ik} = 1,\ \forall i = 1, 2, \cdots, n \end{aligned} \tag{3.1}$$

where $X \in \mathbb{R}^{d\times n}$ is the input data matrix with n images and d-dimensional visual features, $F \in \mathbb{R}^{d\times K}$ is the cluster centroid matrix, and $G \in \mathbb{R}^{n\times K}$ is the cluster assignment matrix and each row of G satisfies the *1-of-K* coding scheme (if data point $\mathbf{x}_i$ is assigned to k-th cluster then $G_{ik} = 1$, and $G_{ik} = 0$, otherwise).

3.2.2 Robust Multi-View K-Means Clustering via Structured Sparsity-Inducing Norm

The original K-means clustering method only works for single-view data clustering. To solve the large-scale multi-view clustering problem, we propose a new multi-view K-means clustering method. Let $X^{(v)} \in \mathbb{R}^{d_v \times n}$ denote the features in v-th view, $F^{(v)} \in \mathbb{R}^{d_v \times K}$ be the centroid matrix for the v-th view, and $G^{(v)} \in \mathbb{R}^{n \times K}$ be the clustering indicator matrix for the v-th view. Given M types of heterogeneous features, $v = 1, 2, \cdots, M$.

The straightforward way to utilize all views of features is to concatenate all features together and perform the clustering algorithm. However, in such method, the important view of features and the less important view of features are treated equally such that the clustering results are not optimal. It is ideal to simultaneously perform the clustering using each view of features and unify their results based their importance to the clustering task. To achieve this goal, we have to solve two challenging problems: 1) how to naturally ensemble the multiple clustering results? 2) how to learn the importance of feature views to the clustering task? More important, we have to solve these issues simultaneously in the clustering objective function, thus previous ensemble approaches cannot be applied here.

When a multi-view clustering algorithm performs clustering using heterogeneous features, the clustering results in different views should be unique, *i.e.* the clustering indicator matrices $G^{(v)}$ of different views should share the same one. Therefore, in multi-view clustering, we force the cluster assignment matrices to be the same across different views, that is, the consensus common cluster indicator matrix $G \in \mathbb{R}^{n \times K}$, which should satisfy the *1-of-K* coding scheme as well.

Meanwhile, as we know, the data outliers greatly affect the performance of K-means clustering, because the K-means solution algorithm is an iterative method and in each iteration we need to calculate the centroid vector. In order to have a more stable clustering performance with respect to a fixed initialization, the robust K-means clustering method is desired. To tackle this problem, we use the sparsity-inducing norm, $\ell_{2,1}$-norm, to replace

the ℓ_2-norm in the clustering objective function, *e.g.* Eq. (3.1). The $\ell_{2,1}$-norm based clustering objective enforces the ℓ_1-norm along the data points direction of data matrix X, and ℓ_2-norm along the features direction. Thus, the effect of outlier data points in clustering are reduced by the ℓ_1-norm. We propose a new robust multi-view K-means clustering method by solving:

$$\begin{aligned} &\min_{F^{(v)},G,\alpha^{(v)}} \sum_{v=1}^{M} (\alpha^{(v)})^{\gamma} ||{X^{(v)}}^T - G{F^{(v)}}^T||_{2,1} \\ &s.t. G_{ik} \in \{0,1\},\ \sum_{k=1}^{K} G_{ik} = 1,\ \sum_{v=1}^{M} \alpha^{(v)} = 1, \end{aligned} \tag{3.2}$$

where $\alpha^{(v)}$ is the weight factor for the v-th view and γ is the parameter to control the weights distribution. We learn the weights for different types of features, such that the important features will get large weights during the multi-view clustering.

3.3 Optimization Algorithm

The difficulty of solving the proposed objective comes from the following two aspects. First of all, the $\ell_{2,1}$-norm is non-smooth. In addition, each entry of the cluster indicator matrix is a binary integer and each row vector must satisfy the *1-of-K* coding scheme. We propose new algorithm to tackle them efficiently.

3.3.1 Algorithm Derivation

To derive the algorithm solving Eq. (3.2), we rewrite Eq. (3.2) as

$$J = \min_{F^{(v)},D^{(v)},\alpha^{(v)},G} \sum_{v=1}^{M} (\alpha^{(v)})^{\gamma} H^{(v)}, \tag{3.3}$$

where

$$H^{(v)} = \text{Tr}\,(X^{(v)} - F^{(v)}G^T)D^{(v)}(X^{(v)} - F^{(v)}G^T)^T. \tag{3.4}$$

$D^{(v)} \in \mathbb{R}^{n\times n}$ is the diagonal matrix corresponding to the v-th view and the i-th entry on the diagonal is defined as:

$$D_{ii}^{(v)} = \frac{1}{2\left\|\mathbf{e}^{(v)i}\right\|}, \quad \forall i = 1, 2, ..., n, \tag{3.5}$$

where $\mathbf{e}^{(v)i}$ is the i-th row of the following matrix:

$$E^{(v)} = X^{(v)^T} - GF^{(v)^T}. \tag{3.6}$$

The first step is fixing G, $D^{(v)}$, $\alpha^{(v)}$ and updating the cluster centroid for each view $F^{(v)}$. Taking derivative of J with respect to $F^{(v)}$, we get

$$\frac{\partial J}{\partial F^{(v)}} = -2X^{(v)}\widetilde{D}^{(v)}G + 2F^{(v)}G^T\widetilde{D}^{(v)}G, \tag{3.7}$$

where

$$\widetilde{D}^{(v)} = (\alpha^{(v)})^\gamma D^{(v)}. \tag{3.8}$$

Setting Eq. (3.7) as 0, we can update $F^{(v)}$:

$$F^{(v)} = X^{(v)}\widetilde{D}^{(v)}G(G^T\widetilde{D}^{(v)}G)^{-1}. \tag{3.9}$$

The second step is fixing $F^{(v)}$, $D^{(v)}$, $\alpha^{(v)}$ and updating the cluster indicator matrix G. We have

$$\begin{aligned} & \sum_{v=1}^{M} \text{Tr}\,(X^{(v)} - F^{(v)}G^T)\widetilde{D}(X^{(v)} - F^{(v)}G^T)^T \\ = & \sum_{v=1}^{M}\sum_{i=1}^{N} \widetilde{D}_{ii}^{(v)}||\mathbf{x}_i^{(v)} - F^{(v)}\mathbf{g}_i||_2^2 \\ = & \sum_{i=1}^{N}(\sum_{v=1}^{M} \widetilde{D}_{ii}^{(v)}||\mathbf{x}_i^{(v)} - F^{(v)}\mathbf{g}_i||_2^2) \end{aligned} \tag{3.10}$$

We can solve the above problem by decoupling the data and assign the cluster indicator for them one by one independently, that is, we need to tackle the following problem for the fixed specific i, with respect to vector $\mathbf{g} = [g_1, g_2, \cdots, g_K]^T \in \mathbb{R}^{K\times 1}$

$$\min_{\mathbf{g}} \sum_{v=1}^{M} \widetilde{d}^{(v)}||\mathbf{x}^{(v)} - F^{(v)}\mathbf{g}||_2^2, \ s.t. g_k \in \{0, 1\}, \ \sum_{k=1}^{K} g_k = 1 \tag{3.11}$$

where $\widetilde{d}^{(v)} = \widetilde{D}_{ii}^{(v)}$ is the i-th element on the diagonal of the matrix $\widetilde{D}^{(v)}$. Given the fact that **g** satisfies *1-of-K* coding scheme, there are K candidates to be the solution of Eq. (3.11), each of which is the k-th column of matrix $I_K = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_K]$. To be specific, we can do an exhaustive search to find out the solution of Eq. (3.11) as,

$$\mathbf{g}^* = \mathbf{e}_k, \tag{3.12}$$

where k is decided as follows,

$$k = \arg\min_j \sum_{v=1}^{M} \widetilde{d}^{(v)} ||\mathbf{x}^{(v)} - F^{(v)}\mathbf{e}_j||_2^2 \ . \tag{3.13}$$

The third step is fixing $F^{(v)}$, G, $\alpha^{(v)}$ and updating $D^{(v)}$ by Eq. (5.26) and Eq. (3.6).

The fourth step is fixing $F^{(v)}$, G, $D^{(v)}$ and updating $\alpha^{(v)}$.

$$\min_{\alpha^{(v)}} \sum_{v=1}^{M} (a^{(v)})^{\gamma} \text{Tr}\ H^{(v)},\ s.t.\ \sum_{v=1}^{M} \alpha^{(v)} = 1,\ \ \alpha^{(v)} \geq 0 \tag{3.14}$$

where $H^{(v)}$ is also defined in Eq. (3.4). Thus, the Lagrange function of Eq. (4.26) is:

$$\sum_{v=1}^{M} (\alpha^{(v)})^{\gamma} H^{(v)} - \lambda(\sum_{v=1}^{M} \alpha^{(v)} - 1). \tag{3.15}$$

In order to get the optimal solution of the above subproblem, set the derivative of Eq. (4.28) with respect to $\alpha^{(v)}$ to zero. We have:

$$\alpha^{(v)} = \left(\frac{\lambda}{\gamma H^{(v)}}\right)^{\frac{1}{\gamma - 1}} \ . \tag{3.16}$$

Substitute the resultant $\alpha^{(v)}$ in Eq. (4.29) into the constraint $\sum_{v=1}^{M} \alpha^{(v)} = 1$, we get:

$$\alpha^{(v)} = \frac{\left(\gamma H^{(v)}\right)^{\frac{1}{1-\gamma}}}{\sum_{v=1}^{M} \left(\gamma H^{(v)}\right)^{\frac{1}{1-\gamma}}} \ . \tag{3.17}$$

By the above four steps, we alternatively update $F^{(v)}$, G, $D^{(v)}$ as well as $\alpha^{(v)}$ and repeat the process iteratively until the objective function becomes converged. We summarize the proposed algorithm in Alg. 5.

3.3.2 Discussion of The Parameter γ

We use one parameter γ to control the distribution of weight factors for different views. From Eq. (4.30), we can see that when $\gamma \to \infty$, we will get equal weight factors. And when $\gamma \to 1$, we will assign 1 to the weight factor of the view whose $H^{(v)}$ value is the smallest and assign 0 to the weights of the other views. Using such a kind of strategy, on one hand, we avoid the trivial solution to the weight distribution of the different views, that is, the solution when $\gamma \to 1$. On the other hand, surprisingly, we can take advantage of only one parameter γ to control the whole weights, reducing the parameters of the model greatly.

3.3.3 Convergence Analysis

We can prove the convergence of the proposed Alg. 5 as follows: We can divide the Eq. (3.2) into four subproblems and each of them is a convex problem with respect to one variable. Therefore, by solving the subproblems alternatively, our proposed algorithm will guarantee that we can find the optimal solution to each subproblem and finally, the algorithm will converge to local solution.

3.4 Time Complexity Analysis

As we know, graph based clustering methods, like spectral clustering and etc., will involve heavy computation, *e.g.* kernel/affinity matrix construction as well as eigen-decomposition. For the data set with n images, the above two calculations will have the time complexity

of $O(n^2)$ and $O(n^3)$ respectively, which makes them impractical for solving the large-scale image clustering problem. Although some research works have been proposed to to reduce the computational cost of the eigen-decomposition of the graph Laplacian [51] [52], they are designed for two-way clustering and have to use the hierarchical scheme to tackle the multi-way clustering problem.

However, our proposed method is centroid based clustering method with the similar time complexity as traditional K-means. For K-means clustering, if the number of iteration is P, then the time complexity is $O(PKnd)$ and the time complexity of our proposed method is $O(PKndM)$, where M is the number of views and usually $P \ll n$, $M \ll n$ and $K \ll n$. In addition, in the real implementation, if the data is too big to store them in memory, we can extend our algorithm as an external memory algorithm that works on a chunk of data at a time and iterate the proposed algorithm on each data chunk in parallel if multiple processors are available. Once all of the data chunks have been processed, the cluster centroid matrix will be updated. Therefore, our proposed method can be used to tackle the very large-scale clustering problem.

Because the graph based multi-view clustering methods cannot be applied to the large-scale image clustering, we did not compare the performance of our method with them in the experiments.

3.5 Experiments

In this section, we will evaluate the performance of the proposed RMKMC method on six benchmark data sets: SensIT Vehicle [53], Caltech-101 [54], Microsoft Research Cambridge Volume 1(MSRC-v1) [55] Handwritten numerals [56], Animal with attribute [57] and SUN 397 [58]. Three standard clustering evaluation metrics are used to measure the

Table 3.1. Data set summary.

Data sets	# of data	# of views	# of cluster
SensIT	300	2	3
Caltech7	441	6	7
MSRC-v1	210	6	7
Digit	2000	6	10
AwA	30475	6	50
SUN	10000	7	100

multi-view clustering performance, that is, Clustering Accuracy (ACC), Normalized Mutual Information(NMI) and Purity.

3.5.1 Data Set Descriptions

We summarize the six data sets that we will use in our experiments in Table 5.1.

SensIT Vehicle data set is the one from wireless distributed sensor networks (WDSN). It utilizes two different sensors, that is, acoustic and seismic sensor to record different signals and do classification for three types of vehicle in an intelligent transportation system. We download the processed data from LIBSVM [59] and randomly sample 100 data for each class. Therefore, we have 300 data samples, 2 views and 3 classes.

Caltech101 data set is an object recognition data set containing 8677 images, belonging to 101 categories. We chose the widely used 7 classes, *i.e.* Faces, Motorbikes, Dolla-Bill, Garfield, Snoopy, Stop-Sign and Windsor-Chair. Following [42], we sample the data and totally we have 441 images. In order to get the different views, we extract LBP [46] with dimension 256, HOG [45] with dimension 100, GIST [47] with dimension 512 and color moment (CMT) [49] with dimension 48, CENTRIST [48] with dimension 1302 and DoG-SIF [44] with dimension 128 visual features from each image.

MSRC-v1 data set is a scene recognition data set containing 8 classes, 240 images in total. Following [41], we select 7 classes composed of tree, building, airplane, cow, face,

car, bicycle and each class has 30 images. We also extract the same 6 visual features from each image with Caltech101 dataset.

Handwritten numerals data set consists of 2000 data points for 0 to 9 ten digit classes. (Each class has 200 data points.) We use the published 6 features to do multi-view clustering. Specifically, these 6 features are 76 Fourier coefficients of the character shapes (FOU), 216 profile correlations (FAC), 64 Karhunen-love coefficients (KAR), 240 pixel averages in 2×3 windows (PIX), 47 Zernike moment (ZER) and 6 morphological (MOR) features.

Animal with attributes is a large-scale data set, which consists of 6 feature, 50 classes, 30475 samples. We utilize all the published features for all the images, that is, Color Histogram (CQ) features , Local Self-Similarity (LSS) features [60], PyramidHOG (PHOG) features [61], SIFT features [44], colorSIFT (RGSIFT) features [62], and SURF features [63].

SUN 397 dataset [58] is a published dataset to provide researchers in computer vision, human perception, cognition and neuroscience, machine learning and data mining, with a comprehensive collection of annotated images covering a large variety of environmental scenes, places and the objects. It consists of 397 classes with 100 images for each class. We conduct the clustering experiment on the top 100 classes via the 7 published features for all the 10000 images.The 7 visual features are color moment, dense SIFT, GIST, HOG, LBP, MAP and TEXTON.

3.5.2 Experimental Setup

We will compare the multi-view clustering performance of our method (RMKMC) with their corresponding single-view counterpart. In addition, we also compare the results of our method with the baseline method naive multi-view K-means clustering (NKMC), and affinity propagation (AP). In our method, when we ignore the weight learning for each type of visual features, the method degenerates to a simple version, called as simple

Table 3.2. SensIT Vehicle data set

Methods	ACC	NMI	Purity
acoustic	0.5049 ± 0.030	0.1018 ± 0.023	0.5055 ± 0.029
seismic	0.5122 ± 0.047	0.1149 ± 0.046	0.5129 ± 0.046
NKMC	0.5449 ± 0.041	0.1375 ± 0.030	0.5465 ± 0.039
AP	0.3867 ± 0.000	0.0084 ± 0.000	0.3867 ± 0.000
SMKMC	0.5490 ± 0.040	0.1395 ± 0.032	0.5494 ± 0.040
RMKMC	**0.5504** ± 0.049	**0.1484** ± 0.033	**0.5542** ± 0.044

MKMC (SMKMC). In order to see the importance of the weight learning, we also compare our method to this simple version method.

Before we do any clustering, for each type of features, we normalize the data first, making all the values in the range $[-1, 1]$. When we implement naive multi-view K-means, we simply use the concatenated normalized features as input for the classic K-means clustering algorithm. As for affinity propagation methods, we need to build the similarity kernel first. Due to the fact that linear kernel is preferred in large-scale problem, we use the following way to construct linear kernel.

$$w_{ij} = \mathbf{x}_i^T \mathbf{x}_j, \quad \forall i, j = 1, 2, ..., n, \tag{3.18}$$

In addition, RMKMC has a parameter γ to control the weight factor distribution among all views. We search the logarithm of the parameter γ, that is, log_{10^γ} in the range from 0.1 to 2 with incremental step 0.2 to get the best parameters γ^*. Since all the clustering algorithms depend on the initializations, we repeat all the methods 50 times using random initialization and report the average performance.

3.5.3 Clustering Results Comparisons

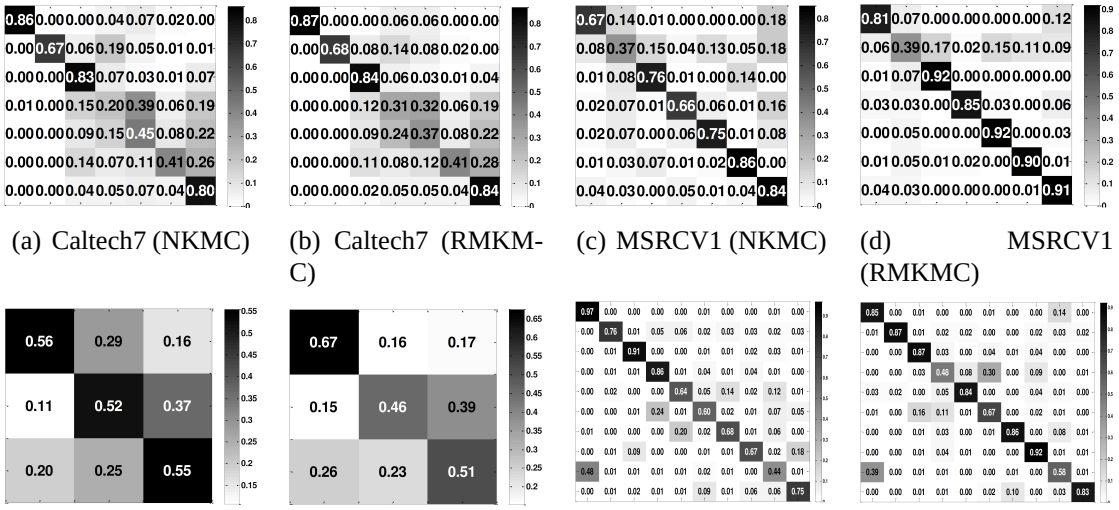
Table 3.2 demonstrates the clustering results on SensIT Vehicle data set. From it, we can see that although there are only two views (acoustic and seismic), compared with

Table 3.3. Caltech101-7 data set.

Methods	ACC	NMI	Purity
LBP	0.5236 ± 0.021	0.4319 ± 0.006	0.6005 ± 0.008
HOG	0.5561 ± 0.052	0.5020 ± 0.035	0.6459 ± 0.038
GIST	0.5663 ± 0.032	0.4737 ± 0.024	0.6418 ± 0.028
CMT	0.3809 ± 0.015	0.2706 ± 0.021	0.4346 ± 0.010
DoG-SIFT	0.6125 ± 0.037	0.5637 ± 0.018	0.6673 ± 0.028
CENTRIST	0.6315 ± 0.058	0.5981 ± 0.046	0.7035 ± 0.044
NKMC	0.6587 ± 0.063	0.6561 ± 0.035	0.7458 ± 0.030
AP	0.5125 ± 0.000	0.3611 ± 0.1054	0.5170 ± 0.1290
SMKMC	0.6723 ± 0.058	0.6775 ± 0.034	0.7561 ± 0.026
RMKMC	**0.6797** ± 0.053	**0.6892** ± 0.029	**0.7595** ± 0.027

single-view K-means counterparts, our proposed RMKMC can boost the clustering performance by more than 10%. Our RMKMC can also beat NKMC and AP. Table 3.3 and Table 3.5 show the clustering results on regular size Caltech101-7, MSRC-v1 as well as Handwritten numerals data set. From it, we can see that with more feature views involved in, our method can improve the clustering performance even further. Also, on large-scale data set Animal with attribute, although doing clustering on a 50 class data set is hard, the performance of our method can still outperform that of the other compared methods as shown in Table 3.6.

We plot the confusion matrices of RMKMC and NKMC in terms of clustering accuracy in Fig. 4.6. Because the clustering numbers of AwA and SUN data sets are large, their confusion matrices cannot be plotted within one page. We skip these two figures. From both tables and figures, we can see that our proposed methods consistently beat the base line method on all the data sets.



(a) Caltech7 (NKMC) (b) Caltech7 (RMKM-C) (c) MSRCV1 (NKMC) (d) MSRCV1 (RMKMC)

(e) sensIT (NKMC) (f) sensIT (RMKMC) (g) digit(NKMC) (h) digit(RMKMC)

Figure 3.1. The calculated average clustering accuracy confusion matrix for Caltech101, MSRCV1, SensIT Vehicle, and Handwritten numerals data sets..

Table 3.4. MSRC-v1 data set.

Methods	ACC	NMI	Purity
LBP	0.4726 ± 0.039	0.4156 ± 0.024	0.5087 ± 0.030
HOG	0.6361 ± 0.041	0.5669 ± 0.032	0.6610 ± 0.037
GIST	0.6283 ± 0.057	0.5523 ± 0.039	0.6511 ± 0.044
CMT	0.5076 ± 0.043	0.4406 ± 0.037	0.5307 ± 0.037
DoG-SIFT	0.4341 ± 0.036	0.3026 ± 0.028	0.4558 ± 0.030
CENTRIST	0.5977 ± 0.062	0.5301 ± 0.037	0.6205 ± 0.054
NKMC	0.7002 ± 0.085	0.6405 ± 0.057	0.7207 ± 0.073
AP	0.1571 ± 0.000	0.2890 ± 0.000	0.1714 ± 0.000
SMKMC	0.7423 ± 0.093	0.6940 ± 0.070	0.7652 ± 0.079
RMKMC	$\mathbf{0.8142} \pm 0.087$	$\mathbf{0.7776} \pm 0.071$	$\mathbf{0.8341} \pm 0.073$

3.6 Conclusion

In this chapter, we proposed a novel robust multi-view K-means clustering methods to tackle the large-scale multi-view clustering problems. Utilizing the common cluster indicator, we can search a consensus pattern and do clustering across multiple visual feature views. Moreover, by imposing the structured sparsity $\ell_{2,1}$-norm on the objective function,

Table 3.5. Handwritten numerals data set.

Methods	ACC	NMI	Purity
FOU	0.5560 ± 0.062	0.5477 ± 0.028	0.5793 ± 0.048
FAC	0.7078 ± 0.065	0.6791 ± 0.032	0.7374 ± 0.051
KAR	0.6898 ± 0.051	0.6662 ± 0.030	0.7149 ± 0.044
MOR	0.6143 ± 0.058	0.6437 ± 0.034	0.6428 ± 0.050
PIX	0.6945 ± 0.067	0.7030 ± 0.040	0.7235 ± 0.059
ZER	0.5348 ± 0.052	0.5123 ± 0.025	0.5684 ± 0.043
NKMC	0.7282 ± 0.067	0.7393 ± 0.039	0.7609 ± 0.059
AP	0.6285 ± 0.000	0.5940 ± 0.000	0.6600 ± 0.000
SMKMC	0.7758 ± 0.079	0.7926 ± 0.039	0.8106 ± 0.060
RMKMC	**0.7889** ± 0.075	**0.8070** ± 0.033	**0.8247** ± 0.052

Table 3.6. Animal with attribute data set.

Methods	ACC	NMI	Purity
CP	0.0675 ± 0.002	0.0773 ± 0.003	0.0874 ± 0.002
LSS	0.0719 ± 0.002	0.0819 ± 0.005	0.0887 ± 0.002
PHOG	0.0690 ± 0.004	0.0691 ± 0.003	0.0823 ± 0.004
RGSIFT	0.0725 ± 0.003	0.0862 ± 0.004	0.0889 ± 0.003
SIFT	0.0732 ± 0.003	0.0944 ± 0.005	0.0919 ± 0.004
SURF	0.0764 ± 0.003	0.0885 ± 0.003	0.0978 ± 0.004
NKMC	0.0802 ± 0.001	0.1075 ± 0.003	0.1007 ± 0.001
AP	0.0769 ± 0.001	0.0793 ± 0.003	0.0975 ± 0.001
SMKMC	0.0841 ± 0.005	0.1108 ± 0.005	0.1039 ± 0.005
RMKMC	**0.0943** ± 0.005	**0.1174** ± 0.005	**0.1140** ± 0.005

our method is robust to the outliers in input data. Our new method learns the weights of each view adaptively. We also introduce an optimization algorithm to iteratively and efficiently solve the proposed non-smooth objective with proved convergence. We evaluate the performance of our methods on six multi-view clustering data sets.

Table 3.7. SUN data set.

Methods	ACC	NMI	Purity
COLOR	0.0507 ± 0.003	0.1417 ± 0.003	0.0544 ± 0.003
DSIFT	0.0661 ± 0.002	0.1717 ± 0.002	0.0710 ± 0.002
GIST	0.0740 ± 0.002	0.2008 ± 0.002	0.0812 ± 0.004
HOG	0.0715 ± 0.003	0.1862 ± 0.003	0.0772 ± 0.003
LBP	0.0599 ± 0.002	0.1618 ± 0.002	0.0644 ± 0.002
MAP	0.0656 ± 0.003	0.1917 ± 0.003	0.0710 ± 0.004
TEXTON	0.0561 ± 0.002	0.1682 ± 0.002	0.0608 ± 0.002
NKMC	0.0546 ± 0.001	0.1507 ± 0.003	0.0591 ± 0.001
AP	0.0667 ± 0.001	0.1693 ± 0.003	0.0765 ± 0.001
SMKMC	0.0834 ± 0.003	0.2106 ± 0.003	0.0839 ± 0.003
RMKMC	**0.0927** ± 0.003	**0.2154** ± 0.003	**0.0922** ± 0.003

Algorithm 5 The algorithm of RMKMC

Input:

1. Data for M views $\{X^{(1)}, \cdots, X^{(M)}\}$ and $X^{(v)} \in \mathbb{R}^{d_v \times n}$.

2. The expected number of clusters K.

3. The parameter γ.

Output:

1. The common cluster indicator matrix G

2. The cluster centroid matrix $F_{(v)}$ for each view.

3. The learned weight $\alpha^{(v)}$ for each view.

Initialization:

1. Set $t = 0$

2. Initialize the common cluster indicator matrix $G \in \mathbb{R}^{n \times K}$ randomly, such that G satisfies the *1-of-K* coding scheme.

3. Initialize the diagonal matrix $D^{(v)} = I_n$ for each view, where $I_n \in \mathbb{R}^{n \times n}$ is the identity matrix.

4. Initialize the weight factor $\alpha^{(v)} = \frac{1}{M}$ for each view.

repeat

1. Calculate the diagonal matrix $\widetilde{D}^{(v)}$ by Eq. (3.8)

2. Update the centroid matrix $F_{(v)}$ for each view by Eq. (3.9)

3. Update the cluster indicator vector **g** for each data one by one via Eq. (3.12) and Eq. (3.13)

4. Update the diagonal matrix $D^{(v)}$ for each view by Eq. (5.26) and Eq. (3.6)

5. Update the weight factor $\alpha^{(v)}$ for each view by Eq. (4.30)

6. Update $t = t + 1$

until Converges

CHAPTER 4

HETEROGENEOUS IMAGE FEATURE INTEGRATION

4.1 Introduction

As we know, scene categorization and visual recognition are key tasks in computer vision research. However, due to images' variability, ambiguity and the wide range of illumination, they are challenging. The most popular way to tackle such problems is to utilize the low-level image features such as global color, texture histograms, object shapes, *etc.* In recent years, a variety of feature representation methods had been proposed to solve how to describe the visual objects in different images. Some focus on the local information, others are holistic descriptors. Among all local feature descriptors, Scale-Invariant Feature Transform (SIFT) [44], Speeded-up Robust Features (SURF) [64], Histogram of Oriented Gradients (HOG) [45] were most popularly used to overcome image variability caused by changing viewpoints, occlusions, and varying illumination. Local Binary Patterns (LBP) was proposed in [46] as a powerful texture feature based on occurrence histogram of local binary patterns. GIST [47] and CENTRIST [48] are two representative holistic descriptors.

Because different features describe different aspects of the visual characteristics, it is true that one descriptor can be regarded as a better representation under certain circumstances than the others. If we integrate all the descriptors via a proper machine learning method, we could create a generally more accurate and more robust descriptor than any single descriptor, which is like the scenario that if we use "multiple view" to observe an object, we can "see" its details more clearly.

How to combine heterogeneous features is becoming a challenging as well as attractive problem nowadays. As a multiple-kernel learning algorithm, the heterogeneous feature

machine (HFM) [65] was recently proposed based on logistic regression loss function and group LASSO regularization to *supervised* fuse the multiple types of features for visual classifications. On the other hand, unsupervised categorization of images or image parts is needed for image and video collection or as a preprocessing step for later supervised classification. In addition, labeling image is a time consuming as well as biased task. Although it is possible to label large amounts of images for research purposes, this is often unrealistic in practice. Therefore, how to take advantage of the heterogeneous features to do unsupervised clustering or semi-supervised learning is still a changing problem.

In this chapter, we will propose two graph based methods to do spectral clustering and semi-supervised learning with the reasonable fusion of heterogeneous modalities , where each modality is a kind of intermediate image descriptor.

4.2 Multi-Modality Spectral Clustering

In recent computer vision research, many unsupervised learning based methods have been proposed to classify scenes and recognize objects from images. Fergus *et al.* [66] and Sivic *et al.* [67] discovered the latent visual building block in images by making use of the generative topic models that were developed for text mining, such as probabilistic Latent Semantic Analysis (pLSA) [68] and Latent Dirichlet Allocation (LDA) [69]. Instead of utilizing the generative models, Grauman *et al.* [70] employed partially matching kernel [71] to get the distinctive model and explored the image category information by spectral clustering. Dueck and Frey applied Affinity Propagation method to cluster different scene and object images [42]. Nevertheless, all these methods only used one image feature descriptor without the help of other descriptors extracted from the same image.

In this section, we unsupervised integrate five renowned descriptors, including DoG-SIFT [44], LBP [46], GIST [47], CENTRIST [48], and HOG [45]. Figure 4.5 demonstrates

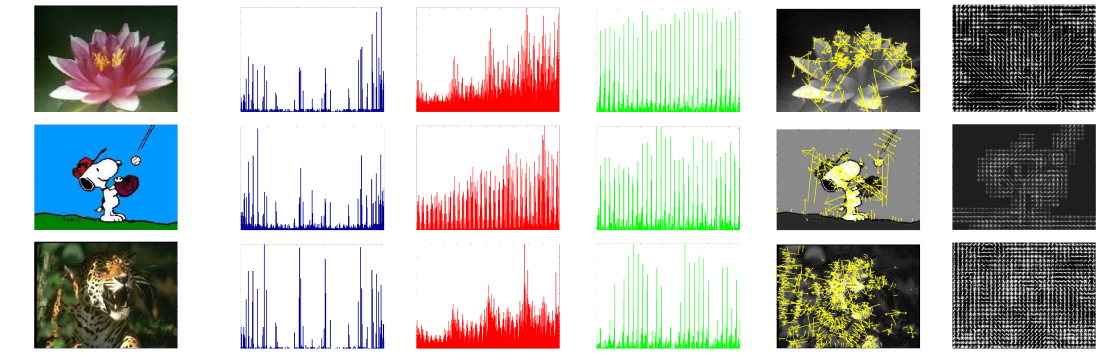


Figure 4.1. The visual patterns of descriptors LBP, GIST, CENTRIST, DoG-SIFT, and HOG of three sample images from Caltech101 data set..

the visual patterns of each descriptor for sample images. Each representation corresponds to a single modal, that is, a local descriptor or holistic descriptor.

4.2.1 Image Descriptors

DoG-SIFT is originally designed for recognizing the same object appearing under different conditions and has been widely used in computer vision and image content retrieval. As a local descriptor, it is invariant to image rotation as well as scale. It is also robust across a substantial range of affine noise and change in illumination. There are several variations of SIFT descriptors (*e.g.* Dense SIFT [72]) in literature. In order to fairly compare our method to existing unsupervised scene categorization methods [42, 70], we use DoG-SIFT to be consistent with their selection.

LBP is a powerful texture feature based on occurrence histogram of local binary patterns. It emphasizes the local structure and is famous for its robustness to rotation and non-uniform illumination.

HOG is a good local descriptor to describe the shape information of the image. Differing to SIFT which describes the feature at the candidate location (keypoint), HOG

describes the feature over the given region. HOG was combined with cell-structured LBP as the human detector and achieves promising performance [73].

GIST encodes rough geometry and spatial structures within an image and suppress detailed texture focusing on the holistic information. It achieved high accuracy in recognizing natural scene categories, *e.g.* mountain and coast. But it often fails to recognize images from indoor environments.

CENTRIST is a holistic descriptor to capture the the stable spatial structure within images that reflects the functionality of the location, and especially suitable for indoor environment categorization classification.

4.2.2 Multi-Modal Spectral Clustering

Generally speaking, there are two main streams for seeking the solutions to multi-modal unsupervised learning problem. One is based on the designed centralized algorithms, making use of the multiple perspectives simultaneously to find out the hidden pattern from the data. The other is to figure out the multi-modal unsupervised clustering problem via a distributed way, that is, to learn the hidden patterns individually from each single representation and then learn the optimal hidden patterns from those multiple patterns [74].

To naturally integrate heterogeneous image features, we propose a unified objective function to simultaneously optimize clustering results of each individual descriptor and their combinations. In other words, we minimize both spectral clustering error of each view and the distances between the multi-modal clustering indicator matrix and each single modal spectral clustering indicator matrix. Therefore, our multi-modal spectral clustering objective function is

$$\min_{G\geq 0, G^T G=I, G_i} \sum_i J(G, G_i), \tag{4.1}$$

where

$$J(G, G_i) = \text{Tr } {G_i}^T L_i G_i + \alpha \text{Tr } (G - G_i)^T (G - G_i) \tag{4.2}$$

where L_i and G_i are the corresponding Laplacian matrix and clustering indicator matrix of each single modal, α is the penalty parameter, G is the multi-modal clustering indicator matrix which we care about. Thus, given the Laplacian matrix of each single modal, we utilize Eq. (4.25) to learn the clustering indicator matrix for each modal and clustering indicator matrix for the multi-modal simultaneously.

4.2.3 Non-Negative Orthonormal Constaint

The traditional way to do spectral clustering results is taking advantage of spectral relaxation. The main disadvantage of this approach is that the obtained spectral solution has mixed signs, which could severely deviate from the true solution and have to resort to other clustering methods, such as K-means or spectral rotation to obtain the final cluster indicators. In order to directly get the discrete cluster indicator matrix without further discretization process, we add the non-negative constraint $G \geq 0$. Compared to the traditional spectral clustering method [75], although we still find the local solution, this relaxation is guaranteed to be converged (will be proved later) and can directly assign clusters to data point. Moreover, it is more robust to the initial conditions.

4.2.4 MMSC Algorithm

In order to get the optimal solution of Eq. (4.25), we set the derivative of the objective function with respect to G_i to zero. We have $2L_iG_i + 2\alpha(G_i - G) = 0$. Thus,

$$G_i = \alpha(L_i + \alpha I)^{-1} G. \tag{4.3}$$

We substitute Eq. (4.3) to Eq. (4.25) and the first term in the summation can be rewritten as:

$$\text{Tr } G_i^T L_i G_i = \alpha^2 \text{Tr } G^T (L_i + \alpha I)^{-1} L_i (L_i + \alpha I)^{-1} G \tag{4.4}$$

Also, since

$$\begin{aligned} & G_i - G \\ &= \alpha (L_i + \alpha I)^{-1} G - G \\ &= (\alpha (L_i + \alpha I)^{-1} - I) G \\ &= (\alpha (L_i + \alpha I)^{-1} - (L_i + \alpha I)(L_i + \alpha I)^{-1}) G \\ &= -L_i (L_i + \alpha I)^{-1} G \end{aligned} \tag{4.5}$$

the second term in the summation can be rewritten as:

$$\begin{aligned} & \alpha \text{Tr } (G_i - G)^T (G_i - G) \\ &= \alpha \text{Tr } G^T (L_i + \alpha I)^{-1} L_i L_i (L_i + \alpha I)^{-1} G \\ &= \alpha \text{Tr } G^T (L_i + \alpha I)^{-1} (L_i + \alpha I) L_i (L_i + \alpha I)^{-1} G \\ &\quad -\alpha^2 \text{Tr } G^T (L_i + \alpha I)^{-1} L_i (L_i + \alpha I)^{-1} G \\ &= \alpha \text{Tr } G^T L_i (L_i + \alpha I)^{-1} G \\ &\quad -\alpha^2 \text{Tr } G^T (L_i + \alpha I)^{-1} L_i (L_i + \alpha I)^{-1} G. \end{aligned} \tag{4.6}$$

We substitute Eq. (4.4) and Eq. (4.6) to Eq. (4.25), and because

$$\begin{aligned} & L_i (L_i + \alpha I)^{-1} \\ &= (L_i + \alpha I - \alpha I)(L_i + \alpha I)^{-1} \\ &= I - \alpha (L_i + \alpha I)^{-1} \end{aligned} \tag{4.7}$$

the optimization problem becomes:

$$\begin{aligned} & \min_{G \geq 0, G^T G = I, G_i} \sum_i \alpha \text{Tr } G^T L_i (L_i + \alpha I)^{-1} G \\ &= \min_{G \geq 0, G^T G = I, G_i} \sum_i \alpha \text{Tr } G^T G - \alpha^2 \text{Tr } G^T (L_i + \alpha I)^{-1} G. \end{aligned} \tag{4.8}$$

Since there is the constrain $G^T G = I$, Eq. (4.8) is equivalent to maximize the following:

$$\max_{G \geq 0, G^T G = I} \text{Tr } G^T (\sum_i (L_i + \alpha I)^{-1}) G. \tag{4.9}$$

The above optimization problem can be solved using an iterative algorithm [76]:

$$G_{ij} \leftarrow G_{ij}\sqrt{\frac{(JG)_{ij}}{(G\beta)_{ij}}}, \qquad \beta \equiv G^T JG, \tag{4.10}$$

where $J = \sum_i (L_i + \alpha I)^{-1}$. We initialize G by $G_0 + 0.2$, where G_0 is obtained by spectral relaxation of Normalized Cut using spectral rotation in the eigenspace. Because G_0 is a cluster indicator matrix, 0.2 is added to make $G_0 + 0.2$ as a valid practical initialization to avoid stucking at the same solution. We can use random initialization as well. However, if we use the above initialization, we can get a more robust clustering result.

Algorithm 6 The algorithm of MMSC

Input: Given V multi-modal affinity matrices W_i, $\forall i = 1, 2, \cdots, V$ and c clusters

Output: Cluster indicator matrix G

Procedure:

1: Calculate the corresponding Laplacian matrices,$L_i = D_i - W_i$, $\forall i = 1, 2, \cdots, V$.

2: Calculate the inverse matrix, $L_{multi-modal} = \sum_i^V (L_i + \alpha I)^{-1}$.

3: Compute the first c eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_c$ of $L_{multi-modal}$.

4: Let $U \in \mathbb{R}^{n \times c}$ be the matrix $U = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_c]$.

5: Calculate the matrix $T \in \mathbb{R}^{n \times c}$ from U by normalizing each row of U to be norm 1.

6: Let $\mathbf{g}^i \in \mathbb{R}^c$, $\forall i = 1, 2, \cdots, n$, be the vector corresponding to the i-th row of T.

7: Cluster the points $(\mathbf{g}^i)_{i=1,2,\cdots,n}$ with spectral rotation algorithm to get G_0

8: Use $G_0 + 0.2$ as the input and perform the iterative algorithm with the non-negative relaxation, to get final assignment indicator matrix G.

4.2.5 Convergence of Our Algorithm

It can be proved that the Eq. (4.10) is guaranteed to converge. First, we write the Lagrangian function of Eq. (4.9) as:

$$\mathcal{L} = \text{Tr}\, G^T J G - \lambda \text{Tr}\, (G^T G - I). \tag{4.11}$$

Theorem 1 *Given the update approach of Eq. (4.10), the lagrangian function $\mathcal{L}$ as in Eq. (4.11) increases monotonically, that is, nondecreasing.*

Proof. We use the auxiliary function [77]. An auxiliary function $P(G, \tilde{G}) \leq \mathcal{L}(G)$ of function $\mathcal{L}(G)$ satisfies

$$P(G, G) = \mathcal{L}(G), \qquad P(G, \tilde{G}) \leq \mathcal{L}(G). \tag{4.12}$$

We define

$$G^{(t+1)} = \arg\max_{G} P(G, G^{(t)}). \tag{4.13}$$

Thus,

$$\mathcal{L}(G^{(t)}) = P(G^{(t)}, G^{(t)}) \leq P(G^{(t+1)}, G^{(t)}) \leq \mathcal{L}(G^{(t+1)}). \tag{4.14}$$

So far, we have shown that $\mathcal{L}(G^{(t)})$ is monotonically increasing. In the following paragraph, we will prove two issues. First, we will prove that we find an appropriate auxiliary function. After that, we will find the global maxima of the auxiliary function. Note that it is important that the maxima in the Eq. (4.13) are the global maxima. Otherwise, the first inequality of Eq. (4.14) does not hold. We can show that

$$\begin{aligned} P(G, \tilde{G}) = & \sum_{k} \sum_{ij} J_{ij} \tilde{G}_{ik} \tilde{G}_{jk} (1 + \log \tfrac{G_{ik} G_{jk}}{\tilde{G}_{ik} \tilde{G}_{jk}}) \\ & - \sum_{i=1}^{p} \sum_{k,l} \tfrac{(\tilde{G}\lambda)_{ik} G_{ik}^2}{\tilde{G}_{ik}} \end{aligned} \tag{4.15}$$

is an auxiliary function of $\mathcal{L}(G)$ of Eq. (4.11) (the constant term λ is ignored). Using the inequality $z \geq 1 + \log z$ and set $z = \frac{G_{ik}G_{jk}}{\tilde{G}_{ik}\tilde{G}_{jk}}$, the first term in Eq. (4.15) is a lower bound of the first term in Eq. (4.11). Since there is a generic inequality

$$\sum_{i=1}^{n}\sum_{p=1}^{k}\frac{(AS'B)_{ip}S_{ip}^2}{S'_{ip}} \geq \text{Tr}\,(S^TASB), \tag{4.16}$$

where $A > 0$, $B > 0$, $S > 0$, $S' > 0$, with A and B symmetric. Taking advantage of that Generic Inequality Eq. (4.19), we can find the second term in Eq. (4.15) is a lower bound of the second term in Eq. (4.11). According to Eq. (4.13), we need to find the global maxima of $P(G, \tilde{G})$ for G. The gradient is

$$\frac{\partial P(G,\tilde{G})}{\partial G_{ik}} = 2\frac{(J\tilde{G})_{ik}\tilde{G}_{ik}}{G_{ik}} - 2\frac{(\tilde{G}\lambda)_{ik}G_{ik}}{\tilde{G}_{ik}}. \tag{4.17}$$

The second derivative is

$$\frac{\partial^2 P(G,\tilde{G})}{\partial G_{ik}\partial G_{jl}} = -2[\frac{(J\tilde{G})_{ik}\tilde{G}_{ik}}{G_{ik}^2} + \frac{(\tilde{G}\lambda)_{ik}}{\tilde{G}_{ik}}]\delta_{ij}\delta_{kl}. \tag{4.18}$$

Therefore, $P(G, \tilde{G})$ is a concave function in H and has a unique global maximum. This global maximum can be obtained by setting the first derivative to zero, which yields

$$G_{ik}^2 = \tilde{G}_{ik}^2\frac{(J\tilde{G})_{ik}}{(\tilde{G}\lambda)_{ik}}. \tag{4.19}$$

According to Eq. (4.13), $G^{(t+1)} = G$ and $G^{(t)} = \tilde{G}$. Thus, we proved the theorem. □

4.2.6 Experimental Results

In this section, we compare the performance of our multi-modal clustering and related methods via two benchmark data sets: Caltech-101 (Fei-Fei et al.2004) as well as Microsoft Research Cambridge Volume 1 (MSRC-v1) (Winn et al.2005). Three standard metrics have been used to measure the image clustering performance, that is, Clustering Accuracy (ACC), Normalized Mutual Information (NMI), and purity.

Table 4.1. Clustering Accuracy

	7 classes	20 classes	MSRC-v1
L	0.4314 ± 0.0065	0.3040 ± 0.0091	0.5613 ± 0.0220
G	0.5439 ± 0.0416	$0.0.4089 \pm 0.0026$	0.6615 ± 0.0051
C	0.60525 ± 0.0185	0.5080 ± 0.0026	0.7258 ± 0.0177
D	0.5766 ± 0.0237	0.2744 ± 0.0057	0.4210 ± 0.0221
H	0.581 ± 0.0542	0.3659 ± 0.0042	0.4966 ± 0.0040
N	0.5137 ± 0.0375	0.3694 ± 0.0049	0.5085 ± 0.0052
K	0.5049 ± 0.0277	0.3383 ± 0.0158	0.5667 ± 0.0518
A	0.5003 ± 0.0000	0.2881 ± 0.0000	0.4476 ± 0.0000
S	0.6327 ± 0.0000	0.2496 ± 0.0000	0.3381 ± 0.0000
M	**0.6244 ± 0.0105**	**0.5237 ± 0.0047**	**0.801 ± 0.0087**

L: LBP, G: GIST, C: CENTRIST, D: DoG-SIFT, H: HOG, N: Naive spectral clustering, K: K-means, A: Affinity Propagation, S: Affnity Propagation with DoG-SIFT, M: MVSC

4.2.6.1 Data Set Descriptions

Caltech-101 Images

The Caltech101 image data set contains 8677 images of objects, each with approximately 0.1 mega pixel resolution, belonging to 101 categories. We follow [42] to choose 7 and 20 classes data set respectively from 101 classes. The 7 classes include Faces, Motorbikes, Dolla-Bill, Garfield, Snoopy, Stop-Sign, Windsor-Chair and have 441 images in total. The 20 classes include Faces, Leopards, Motorbikes, Binocular, Brain, Camera, Car-Side, Dollar-Bill, Ferry, Garfield, Hedgehog, Pagoda, Rhino, Snoopy, Stapler, Stop-Sign, Water-Lilly, Windsor-Chair, Wrench, Yin-Yang and have 1230 images all together.

MSRC-v1 Images

We follow Lee and Grauman's approach [41] to refine the data set, getting 7 classes composed of tree, building, airplane, cow, face, car, bicycle, and each refined class has 30 images. Compared to the Caltech-101 data set, MSRC-v1 has more clutter and variability in the objects appearances.

Table 4.2. Normalized Mutual Information.

	7 classes	20 classes	MSRC-v1
L	0.4177 ± 0.0104	0.3807 ± 0.0047	0.4411 ± 0.0129
G	0.5443 ± 0.0199	0.4846 ± 0.0041	0.6322 ± 0.0236
C	0.5284 ± 0.0136	0.5503 ± 0.0025	0.5966 ± 0.0163
D	0.5930 ± 0.0110	0.2873 ± 0.0052	0.2613 ± 0.0160
H	0.4748 ± 0.0225	0.4326 ± 0.0030	0.4318 ± 0.0023
N	0.4828 ± 0.0027	0.4337 ± 0.0030	0.4560 ± 0.0133
K	0.5298 ± 0.0463	0.4004 ± 0.0130	0.4803 ± 0.0384
A	0.4807 ± 0.0000	0.3766 ± 0.0000	0.5376 ± 0.0000
S	0.5139 ± 0.0000	0.3242 ± 0.0000	0.4798 ± 0.0000
M	$\mathbf{0.6865 \pm 0.0053}$	$\mathbf{0.5915 \pm 0.0039}$	$\mathbf{0.7405 \pm 0.0127}$

L: LBP, G: GIST, C: CENTRIST, D: DoG-SIFT, H: HOG, N: Naive spectral clustering, K: K-means, A: Affinity Propogation, S: Affnity Propagation with DoG-SIFT, M: MVSC

4.2.6.2 Experimental Setup

We extract LBP, GIST, CENTRIST, DoG-SIFT, and HOG descriptors respectively from each image and use the Gaussian Kernel to get the similarity matrices for LBP, GIST, CENTRIST and HOG. In order to solve the inequality length problem of the DoG-SIFT feature, we resort to the pyramid match kernel [71] to build the similarity matrix, using the LIBPMK toolkit. Thus, given an image, we have five similarity (affinity) matrices calculated from five different features. Regarding the parameter σ for Gaussian Kernel, we resort to the self tuning method [78].

We apply the spectral clustering algorithm [75] to do the clustering with each single modal method. Within these five methods (corresponding five modals), the spectral clustering plus DoG-SIFT is the method used in [70]. We also implement Affinity Propagation plus DoG-SIFT method that was proposed in [42].

In order to further show the power of our MMSC method, we concatenate these five features to get a large feature vector and use Gaussian Kernel to calculate a unified similarity matrix. We also evaluate the clustering performances of classical spectral clustering,

Table 4.3. Clustering Purity.

	7 classes	20 classes	MSRC-v1
L	0.5727 ± 0.0049	0.3607 ± 0.0070	0.5622 ± 0.0195
G	0.6683 ± 0.0237	0.4711 ± 0.0045	0.7084 ± 0.0228
C	0.6942 ± 0.0165	0.5554 ± 0.0026	0.7258 ± 0.0177
D	0.7016 ± 0.0203	0.3114 ± 0.0067	0.4390 ± 0.0209
H	0.5921 ± 0.0154	0.4227 ± 0.0031	0.5537 ± 0.0040
N	0.5968 ± 0.0144	0.4248 ± 0.0028	0.5614 ± 0.0128
K	0.6507 ± 0.0455	0.3713 ± 0.0167	0.5882 ± 0.0474
A	0.5941 ± 0.0000	0.3691 ± 0.0000	0.5857 ± 0.0000
S	0.6372 ± 0.0000	0.3967 ± 0.0000	0.5619 ± 0.0000
M	$\mathbf{0.7639 \pm 0.0009}$	$\mathbf{0.5777 \pm 0.0009}$	$\mathbf{0.8048 \pm 0.0085}$

L: LBP, G: GIST, C: CENTRIST, D: DoG-SIFT, H: HOG, N: Naive spectral clustering, K: K-means, A: Affinity Propogation, S: Affnity Propagation with DoG-SIFT, M: MVSC

K-means and Affinity Propagation on this new similarity matrix. Thus, we compare our MMSC approach to total nine existing methods.

As we know, the results of all clustering algorithms depend on the initial conditions. Therefore, we average 50 iterations to get the average and standard deviations of three evaluation metrics for each method and fix the penalty parameter $log_{10}\alpha$ in the range from -2 to 2 with incremental step 0.2.

4.2.6.3 Clustering Results Comparison

The results are shown in Table 4.1, Table 4.2.6.1, Table 4.2.6.1. we can conclude that utilizing our MVSC algorithm, we can always obtain a better clustering quality at least 5 percent than the single view or other state-of-the-art unsupervised image categorization methods.

For demonstration purpose, we randomly pickup the Garfield image class from Caltech101 7-class data set and show top 28 nearest images to the cluster centroid (gotten by applying spectral clustering to each single view and our MVSC method) in Fig. 4.4. Obvi-

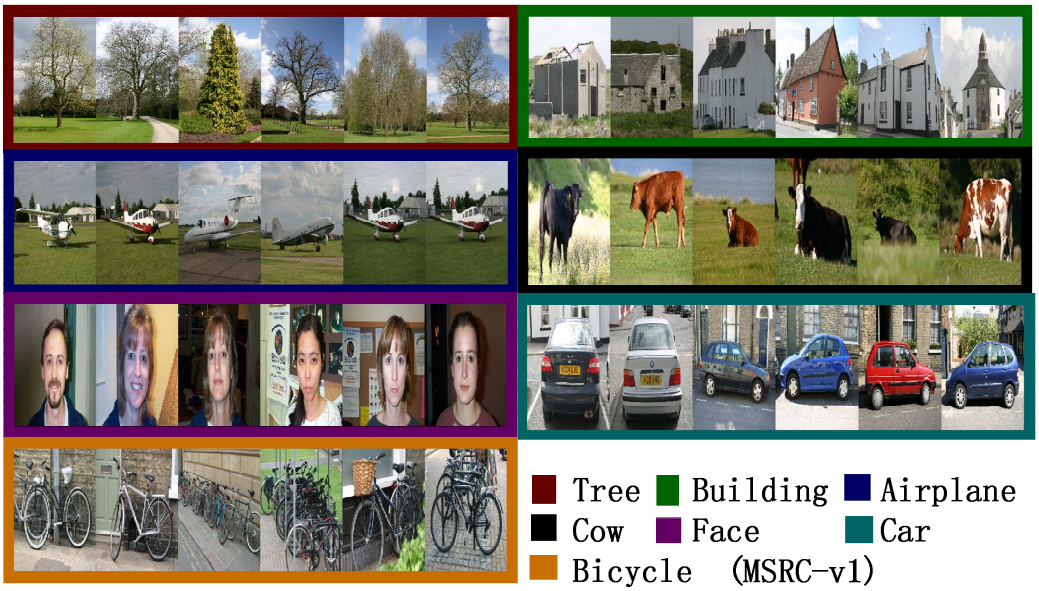


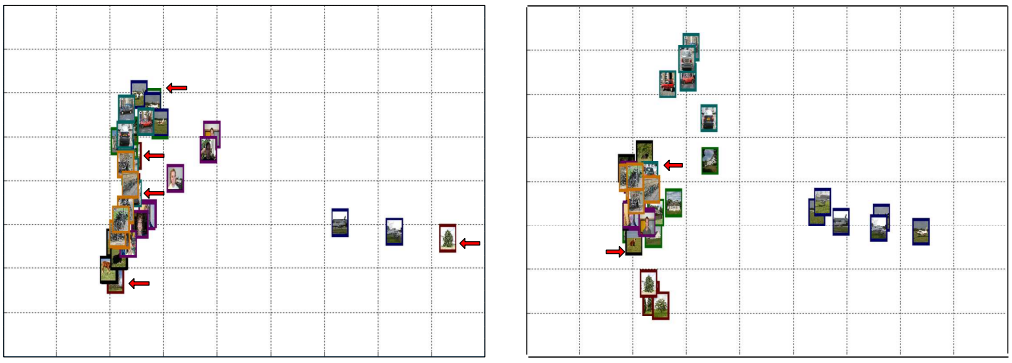
Figure 4.2. The randomly selected image samples for MSRCv-1 data.

ously the integration of different views can cluster more Garfield images into correct group than each individual view.

From MSRC-v1 data, we randomly select 20% images from each class, and Fig. 4.2 shows the selected images. We project them to the 2nd and 3rd eigenvectors of graph Laplacian matrices of each individual view and multi-view. Fig. 4.3 illustrates the projection results. Obviously the performance of feature fusion by MMSC method is the best. Note that we use red arrows to point to the images which are visually far away from other images in the same group, *i.e.* wrong clustering results. Because MSRC-v1 data have 7 classes and each class has 30 images, we cannot plot all of them on the figure (otherwise, many of them will be overlapped each other).

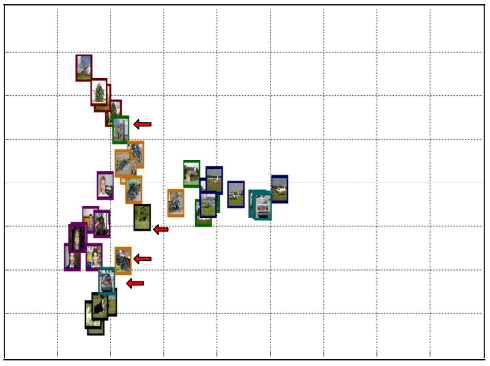
4.2.6.4 Visual Analysis

In MSRC-v1 data, because lots of tree (red frame), building (green frame), cow (black frame), and airplane (dark blue) images have large grass background area, if we only

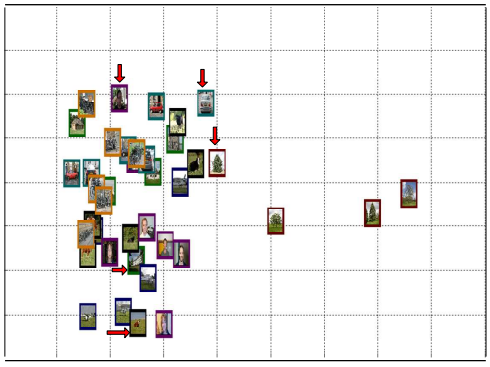


(a) LBP

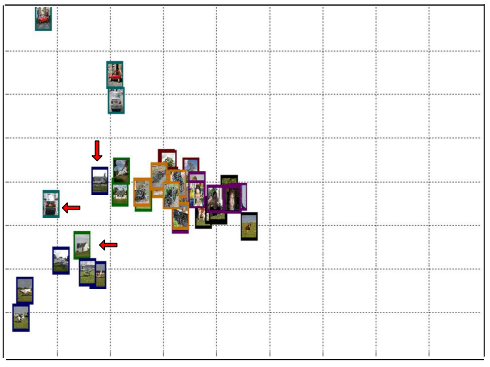
(b) GIST



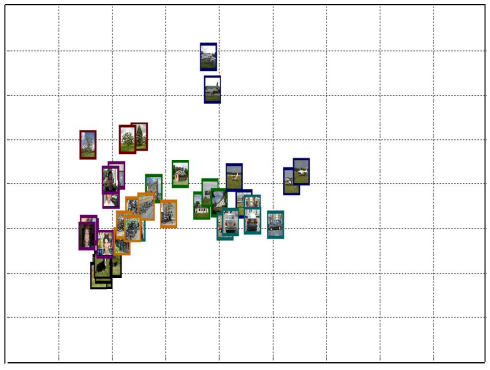
(c) CENTRIST



(d) DoG-SIFT



(e) HOG



(f) MMSC

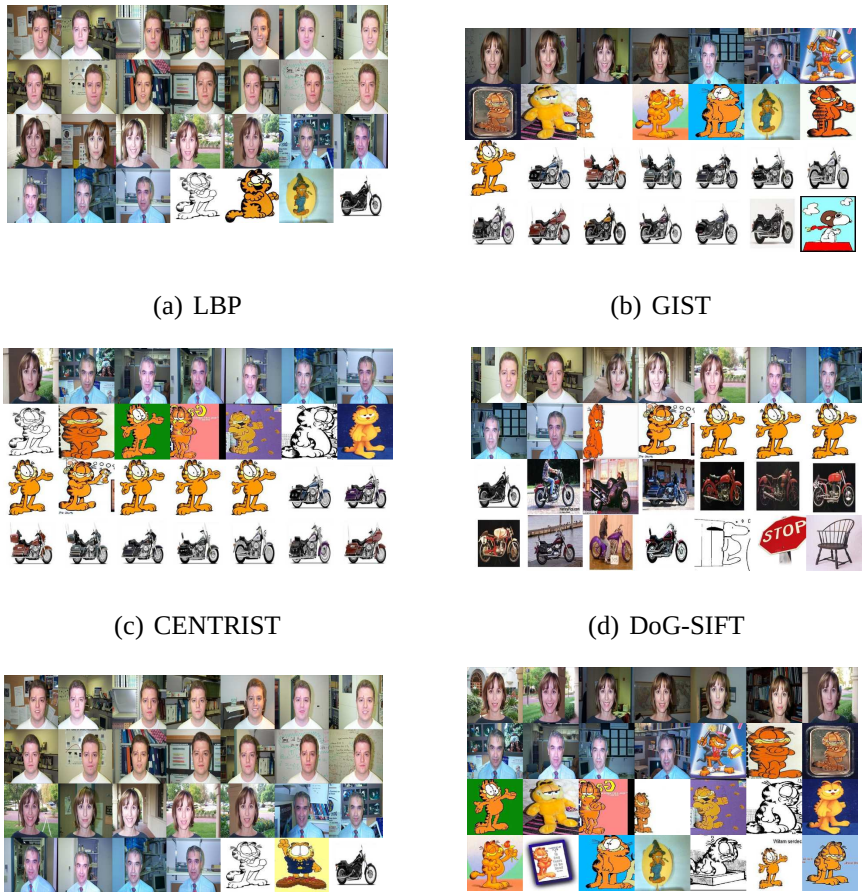
Figure 4.3. The visual clustering performance of MMSC projected to the 2nd and 3rd eigen-vector plane for MSRC-v1 data set..

use the local descriptors, lots of features are prone to falling into the background area and these descriptors in the background area may “look” very similar. Our results have shown that the clustering performance using local descriptors (LBP, DoG-SIFT, HOG) for such four categories images is worse than that using holistic descriptors (GIST, CENTRIST) as shown in Fig. 4.3. However, if we choose holistic descriptor only, we cannot achieve good cluster performance neither. Because we will ignore many useful detail information, which is like the case that we are prone to confusing car images shown in cyan frame with bicycle images shown in orange frame in Fig. 4.3(b) and Fig. 4.3(c). Another interesting thing is that from our results, we observe if we combine the features not properly, the performance of using one large feature vector can be worse than that using only one feature, even by classical clustering methods, like Naive spectral clustering, K-means, and Affinity propagation, which again demonstrates the power of our MMSC algorithm.

In Caltech 101 data, for both 7 classes and 20 classes, the majority of images have varying degrees of background clutter, which will affect the clustering results. From Fig. 4.4, we can see that Garfield with uniform background will be clustered with motorbike images with higher probability using holistic descriptors (GIST and CENTRIST). Moreover, since the shape of Garfield’s face and human’s face are similar: round shape, two eyes, one nose and one mouth, the descriptors focusing on local shape information (LBP and HOG) will cluster more face images with garfield images as well.

4.3 Heterogeneous Image Features Integration via Multi-Modal Semi-Supervised Learning Model

As we know, in the traditional supervised learning paradigm, increasing the quantity and diversity of labeled images enhances the performance of the learned classifier. Nevertheless, labeling image is a time consuming as well as biased task. Although it is possible



(a) LBP (b) GIST

(c) CENTRIST (d) DoG-SIFT

(e) HOG (f) MMSC

Figure 4.4. Clustering results of different methods on Garfield cluster in Caltech 101 data set. The top 28 nearest images to the centroid are visualized..

to label large amounts of images for research purposes, this is often unrealistic in practice. To solve the classification problem caused by the scarce or expensive labeled data, we resort to semi-supervised learning, which takes advantage of the combination of both labeled and unlabeled images.

The most popular way to do semi-supervised learning for image categorization is to use some low-level image descriptors. In order to overcome the image content representation issue, more and more visual descriptors have been proposed. Some focus on the local information, while others are holistic descriptors. If we integrate all the descriptors

via a proper learning method, we could create a generally more accurate and more robust descriptor than any single one.

In this section, we propose a novel semi-supervised learning approach to integrate heterogeneous features from both labeled and unlabeled as well as unsegmented images. Considering each type of feature as one modality, taking advantage of the large amount of unlabeled data information, our new adaptive multi-modal semi-supervised classification (AMMSS) algorithm propagates the class labels from labeled images to unlabeled images based on the integrated multi-modal feature similarity and learn the weights for different modalities (image features) simultaneously. We applied our AMMSS method to integrate multiple popularly used image features, which describe the image content from different perspectives, and evaluated the performance by four benchmark datasets. Compared with the existing semi-supervised scene and object categorization methods, our approach always achieves superior performances in terms of both macro and micro classification accuracy.

4.3.1 Basic Framework of Graph Based Semi-Supervised Learning

Assume we have n images $X = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, where each image is abstracted as a data point $\mathbf{x}_i \in \mathbb{R}^p$. Each data point $\mathbf{x}_i$ belongs to one of K classes $C = \{c_1, \cdots, c_K\}$ represented by $\mathbf{y}_i \in \{0,1\}^K$, such that $\mathbf{y}_i(k) = 1$ if $\mathbf{x}_i$ is classified into k-th class, and 0 otherwise. Without loss of generality, we assume the first $l \ll n$ data are already labeled, which are denoted as $T = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{l}$. Our task is to learn a function $f : X \to \{0,1\}^K$ from T that is able to classify the given unlabeled data $\mathbf{x}_i (l+1 \leq i \leq n)$ into one and only one class in C. For simplicity, we use u to denote the number of unlabeled data point. that is, $l + u = n$ and split the label matrix $Y = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n]^T$, $\mathbf{y}_i \in \mathbb{R}^K$ into 2 blocks: $Y = \begin{bmatrix} Y_l \\ Y_u \end{bmatrix}$.

Given the dataset X, all the image data including the labeled and unlabeled ones are abstracted as the vertices on $K-NN$ graph. To be specific, we connect $\mathbf{x}_i$, $\mathbf{x}_j$ if one of them is among the other's K-nearest neighbor by Euclidean distance and define the corresponding weight on the edge as the following,

$$w_{ij} = \begin{cases} \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}), & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ are connected} \\ 0, & \text{otherwise} \end{cases} \tag{4.20}$$

where σ is the bandwidth parameter. Therefore, $W = \{w_{i,j}\}$ is an $(l+u) \times (l+u)$ symmetric undirected matrix with non-negative edge weight. Let $d_{ii} = \sum_{j=1}^{l+u} w_{ij}$ and D be the diagonal matrix by substituting d_{ii}, $i = 1, 2, ...(l+u)$ on the diagonal. The normalized graph Laplacian matrix L is defined as

$$L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{4.21}$$

4.3.2 Label Propogation for Single Modality

According to graph theory, if the edge weight between two vertices on affinity matrix is large, then the class labels of these two instances should be similar. Based on the above assumption, denote $G \in \mathbb{R}^{n \times K}$ as the class label matrix, for each feature modality, we use the following way to propagate the class label information from labeled data to unlabeled data,

$$\min_G G^T L G \quad s.t. \quad \mathbf{g}_i = \mathbf{y}_i, \quad \forall i = 1, 2, ..., l, \tag{4.22}$$

where L is the normalized Laplacian matrix defined in Eq. (4.21).

Eq. (4.22) can be rewritten as the following,

$$\min_{G_u} \text{Tr} \left(\begin{bmatrix} Y_l \\ G_u \end{bmatrix}^T \begin{bmatrix} L_{ll} & L_{lu} \\ L_{ul} & L_{uu} \end{bmatrix} \begin{bmatrix} Y_l \\ G_u \end{bmatrix} \right), \tag{4.23}$$

since we know the labels Y_l for the first l instances, which has the following unique solution,

$$G_u = -L_{uu}^{-1} L_{ul} Y_l \tag{4.24}$$

4.3.3 Label Propogation by AMMSS

In order to properly and naturally integrate heterogeneous image features to do semi-supervised learning, we need a co-regularization term to learn a consensus class label matrix and let the differences between that consensus label matrix and the class label matrix of each feature modality as small as possible. With the addition of weight factor for each feature modality, we adaptively learn the weight for each feature modality, assigning the more discriminative modality with higher weight. We summarize the proposed AMMSS method as the following objective function,

$$\begin{aligned} &\min_{G,G^{(v)},\alpha^{(v)}} \sum_{v=1}^{V} (\alpha^{(v)})^r \text{Tr}\,(G^{(v)^T} L^{(v)} G^{(v)}) \\ &+\lambda \sum_{v=1}^{V} \text{Tr}\,((G - G^{(v)})^T (G - G^{(v)})) \\ &s.t. \quad \mathbf{g}_i = \mathbf{y}_i, \quad \forall i = 1, 2, ...l, \qquad \sum_{v=1}^{V} \alpha^{(v)} = 1, \\ &\quad \alpha^{(v)} \geq 0, \end{aligned} \tag{4.25}$$

where V is the number of image visual features, $\alpha^{(v)}$ is the non-negative normalized weight factor for the v-th modality, $L^{(v)}$ and $G^{(v)}$ are the normalized Laplacian matrix and class label matrix for the v-th feature modality respectively. G is the shared consensus class label matrix that we are interested. We use the scalar r to control the distribution of different weights for different feature modalities and λ is the regularization parameter to balance the 1st term and the 2nd term. We want to solve for G, $G^{(v)}$ and $\alpha^{(v)}$ simultaneously via the proposed Eq. (4.25).

4.3.4 Optimization Algorithms

4.3.4.1 The Optimization Algorithm of AMMSS

We decompose Eq. (4.25) as the following three subproblems and solve them alternatively and iteratively.

The first step is fixing G and $G^{(v)}$, solving $\alpha^{(v)}$. Then, the objective function becomes

$$\begin{aligned} &\min_{\alpha^{(v)}} \sum_{v=1}^{V} (\alpha^{(v)})^r \text{Tr}\,(G^{(v)^T} L^{(v)} G^{(v)}), \\ &s.t. \sum_{v=1}^{V} \alpha^{(v)} = 1,\ \alpha^{(v)} \geq 0 \end{aligned} \tag{4.26}$$

Let $p^{(v)} = \text{Tr}\,(G^{(v)^T} L^{(v)} G^{(v)})$, then the Eq. (4.26) can be rewritten as

$$\sum_{v=1}^{V} (\alpha^{(v)})^r p^{(v)}, \quad s.t. \quad \sum_{v=1}^{V} \alpha^{(v)} = 1, \quad \alpha^{(v)} \geq 0 \tag{4.27}$$

Thus, the Lagrange function of Eq. (4.27) is

$$\sum_{v=1}^{V} (\alpha^{(v)})^r p^{(v)} - \beta(\sum_{v=1}^{V} \alpha^{(v)} - 1) \tag{4.28}$$

where β is the Lagrange multiplier. In order to get the optimal solution of the above subproblem, set the derivative of Eq. (4.28) with respect to $\alpha^{(v)}$ to zero. We have

$$\alpha^{(v)} = \left(\frac{\beta}{r p^{(v)}}\right)^{\frac{1}{r-1}} \tag{4.29}$$

Substitute the resultant $\alpha^{(v)}$ in Eq. (4.29) into the constraint $\sum_v \alpha^{(v)} = 1$, we get

$$a^{(v)} = (r p^{(v)})^{\frac{1}{1-r}} / \sum_{v=1}^{V} (r p^{(v)})^{\frac{1}{1-r}} \tag{4.30}$$

The second step is fixing $\alpha^{(v)}$ and G, solving $G^{(v)}$. We change the variable and let $\tilde{L}^{(v)} = (\alpha^{(v)})^r L^{(v)}$ then the objective function becomes

$$\begin{aligned} &\min_{G, G^{(v)}} \sum_{v} \text{Tr}\,(G^{(v)^T} \tilde{L}^{(v)} G^{(v)}) \\ &+ \qquad \lambda \sum_{v} \text{Tr}\,((G - G^{(v)})^T (G - G^{(v)})) \\ &s.t. \quad \mathbf{g}_i = \mathbf{y}_i, \quad \forall i = 1, 2, ..., l \end{aligned} \tag{4.31}$$

Set the derivative of Eq. (4.31) with respect to $G^{(v)}$ to zero. We have

$$G^{(v)} = \lambda(\tilde{L}^{(v)} + \lambda I)^{-1} G \tag{4.32}$$

The third step is fixing $\alpha^{(v)}$ and $G^{(v)}$, solving G. Substitute the resultant $G^{(v)}$ in Eq. (4.32) into the Eq. (4.31), we get (The proof is in Appendix)

$$\begin{aligned} &\sum_v \text{Tr}\,(G^{(v)^T} \tilde{L}^{(v)} G^{(v)}) \\ &\quad + \lambda \sum_v \text{Tr}\,((G - G^{(v)})^T (G - G^{(v)})) \\ &= \lambda \text{Tr}\,(G^T (\sum_v (I - \lambda(\tilde{L}^{(v)} + \lambda I)^{-1}) G) \end{aligned} \tag{4.33}$$

Let $H = \sum_v \left(I - \lambda(\tilde{L}^{(v)} + \lambda I)^{-1} \right)$. Therefore, Eq. (4.31) is equivalent to the following optimization problem,

$$\begin{aligned} &\min_G \text{Tr}\,(G^T H G) \\ &s.t. \quad \mathbf{g}_i = \mathbf{y}_i, i = 1, 2, ..., l \end{aligned} \tag{4.34}$$

To compute class label matrix for the unlabeled image explicitly in terms of matrix operations, we split the matrix H into 4 blocks by the l-th row and l-th column:

$$H = \begin{bmatrix} H_{ll} & H_{lu} \\ H_{ul} & H_{uu} \end{bmatrix} \tag{4.35}$$

Therefore,

$$\begin{aligned} &\text{Tr}\,(G^T H G) \\ &= \text{Tr}\left(\begin{bmatrix} G_l \\ G_u \end{bmatrix}^T \begin{bmatrix} H_{ll} & H_{lu} \\ H_{ul} & H_{uu} \end{bmatrix} \begin{bmatrix} G_l \\ G_u \end{bmatrix} \right) \\ &= \text{Tr}\left(\begin{bmatrix} Y_l \\ G_u \end{bmatrix}^T \begin{bmatrix} H_{ll} & H_{lu} \\ H_{ul} & H_{uu} \end{bmatrix} \begin{bmatrix} Y_l \\ G_u \end{bmatrix} \right) \\ &= \text{Tr}\,(Y_l^T H_{ll} Y_l + G_u^T H_{ul} Y_l + Y_l^T H_{lu} G_u + G_u^T H_{uu} G_u) \\ &= \text{Tr}\,(Y_l^T H_{ll} Y_l + G_u^T H_{ul} Y_l + G_u^T H_{ul} Y_l + G_u^T H_{uu} G_u) \end{aligned} \tag{4.36}$$

Thus optimization problem in Eq. (4.34) is equivalent to the subsequent problem,

$$\min_{G_u}[2\text{Tr}\,(G_u^T H_{ul} Y_l) + \text{Tr}\,(G_u^T H_{uu} G_u)] \tag{4.37}$$

Setting the derivative of Eq. (4.37) to zero with respect to G_u, we get

$$G_u = -H_{uu}^{-1} H_{ul} Y_l \tag{4.38}$$

By the above three steps, we alternatively update $\alpha^{(v)}$, $G^{(v)}$ and G and repeat them iteratively until the objective function converges. At last, we resort to the following decision function to assign the single class label to the unlabeled images,

$$\mathbf{y}_i = \arg\max_j G_{ij},\ \forall i = l+1, l+2, ..., n.\ \forall j = 1, 2, ..., K. \tag{4.39}$$

We summarize the algorithm in Alg. 7.

4.3.4.2 Convergence of The Algorithm

We will prove the convergence of the proposed Alg. 7 as following: We divide the original problem Eq. (7) into three subproblems and each of them is convex problem. Since the original problem is not a joint convex problem, by solving the subproblems alternatively, Alg. 7 will converge to the local solution and we use $1/V$ as the initial weight for each modality. Later in our experiment we will demonstrate the fast convergence of our algorithm.

4.3.4.3 Discussion of The Parameter r

In AMMSS, we use one parameter r to control the distribution of weight factors for different feature modalities. From Eq. (4.30), we can see that when $r \to \infty$, we will get equal weight factors. And when $r \to 1$, we will assign 1 to the weight factor of the modality whose $p^{(v)}$ value is the smallest and assign 0 to the weights of other modalities. Using such

kind of strategy, on one hand, we avoid the trivial solution to the weight distribution of the different modalities, that is, the solution when $r \to 1$. On the other hand, surprisingly, we can take advantage of only one parameter r to control the whole weights, reducing the parameters of the model greatly.

4.3.5 Experimental Results

Since our AMMSS is a kind of graph based semi-supervised learning algorithm, we will compare the performance of our AMMSS and related graph based state-of-art semi-supervised methods on five benchmark datasets: Caltech-101 [54], Microsoft Research Cambridge Volume 1 (MSRC-v1) [55], Handwritten numerals (HW) [56] and Animal with Attributes(AwA) [57]. The image classification performance is evaluated in terms of average macro and micro classification accuracy.

4.3.5.1 Dataset Descriptions

Caltech-101 Images The Caltech101 image dataset contains 8677 images of objects, each with approximately 0.1 mega pixel resolution, belonging to 101 categories. We follow [42] to choose 7 and 20 classes dataset respectively from 101 classes. The 7 classes include Faces, Motorbikes, Dolla-Bill, Garfield, Snoopy, Stop-Sign, Windsor-Chair and have 441 images in total. The 20 classes include Faces, Leopards, Motorbikes, Binocular, Brain, Camera, Car-Side, Dollar-Bill, Ferry, Garfield, Hedgehog, Pagoda, Rhino, Snoopy, Stapler, Stop-Sign, Water-Lilly, Windsor-Chair, Wrench, Yin-Yang and have 1230 images all together.

MSRC-v1 Images

We follow Lee and Grauman's approach [41] to refine the dataset, getting 7 classes composed of tree, building, airplane, cow, face, car, bicycle, and each refined class has 30 images. Compared to the Caltech101 dataset, MSRC-v1 has more clutter and variability in

the objects appearances. Since there is no published image descriptors for Caltech-101 and MSRC-v1 datasets, we extract the following six popular visual features for each image: On one hand, we extract three holistic visual features for each image, *i.e.* 45 dimension color moment (CMT) [49]; 512 dimension GIST feature [47]; 1302 dimension CENTRIST feature [48]. On the other hand, we collect three local descriptor as well, *i.e.* 256 dimension local binary pattern (LBP) [46]; 576 dimension HOG feature and famous 128 dimension DoG-SIFT descriptor [44].

Handwritten numerals (HW)

Handwritten numerals dataset consists of 2000 data point for 0 to 9 ten digit classes. (Each class has 200 data points.) We use the published six visual features [56] extracted from each image. Specifically, the six visual features are 76 dimension Fourier coefficients of the character shapes (FOU), 216 dimension profile correlations (FAC), 64 dimension Karhunen-love coefficients (KAR), 240 dimension pixel averages in 2×3 windows (PIX), 47 dimension Zernike moment (ZER) and 6 dimension morphological (MOR) features.

Animal with attributes (AWA)

Animal with attributes data set is the largest data set, which is also an image data set consisting of 6 feature 50 classes. We randomly sample 50 images for each class and get 2500 images in total. We utilize all the published features, that is, 2688 dimension Color Histogram (CQ) features , 2000 dimension Local Self-Similarity (LSS) features , 252 dimension PyramidHOG (PHOG) features, 2000 dimension SIFT features, 2000 dimension colorSIFT (RGSIFT) feature and 2000 dimension SURF features.

4.3.5.2 Experimental Setup

We use the Gaussian Kernel in Eq. (4.20) with 7-nearest neighbor to get the affinity matrices for different visual features. We utilize self-tuning method [78] to calculate the bandwidth parameter σ. In order to solve the inequality length problem of the DoG-SIFT

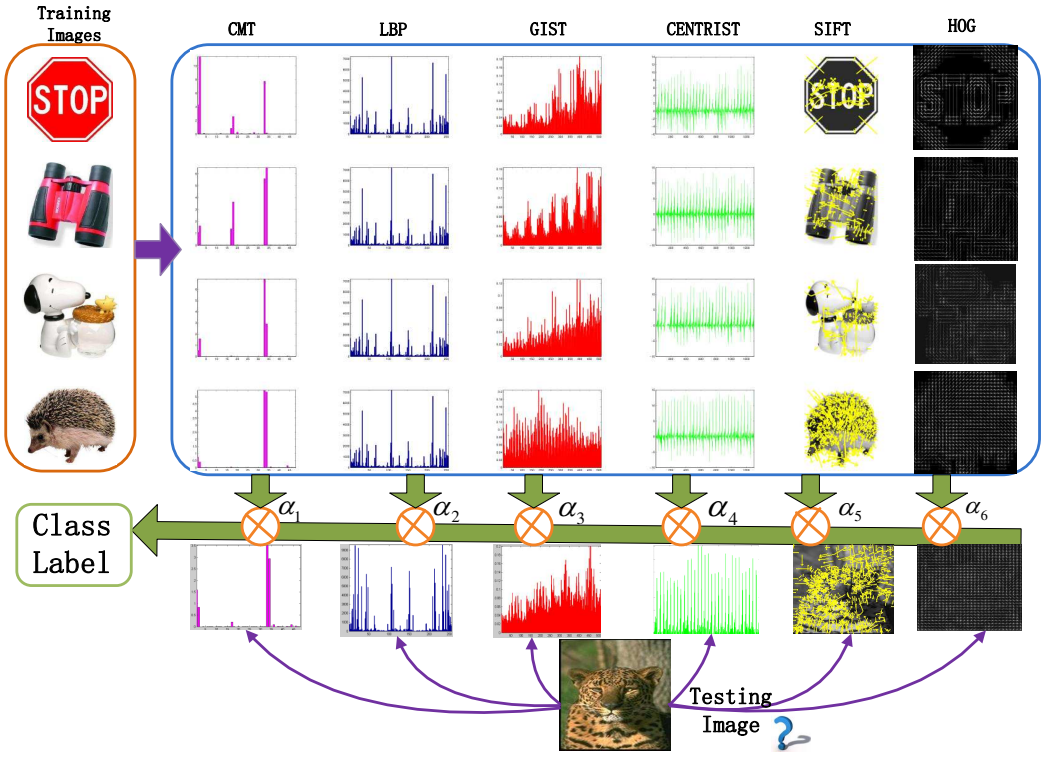


Figure 4.5. The demonstration of different visual descriptors from Caltech 101 dataset. The final class label of the testing image is decided by the weighted six different feature modalities, where the weight for different feature modality is learned by the training images..

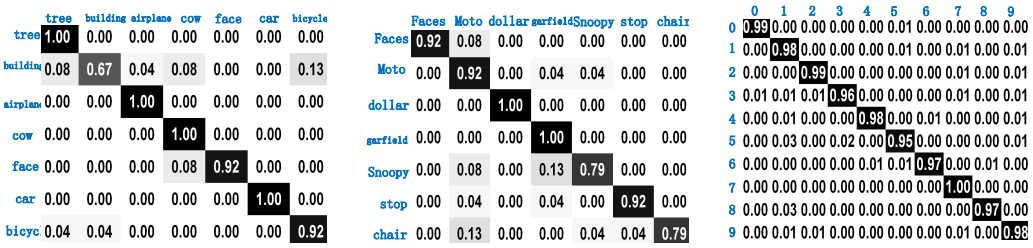
feature, we utilize the pyramid match kernel [71] to build the similarity matrix, using the LIBPMK toolkit. Thus, given an image, we have multiple similarity (affinity) matrices calculated from different modalities. In our experiment for each dataset to mimic the "real" situation in semi-supervised learning case ($l \ll u$), we randomly choose 20% data for training and use the rest for testing. We repeat the above procedure 10 times and report the average result. r is the parameter to control the distribution of the weights for different feature modalities, which we will discuss in detail later. We search the logarithm of the parameter r, that is, $log_{10}r$ in the range from 0.1 to 2 with incremental step 0.2 and search the regularization parameter λ in the range from 0 to 1 with incremental step 0.1 to get the

best parameters r^* as well as λ^* based on the 2-fold cross validation inside the training data only.

4.3.5.3 Classification Results Comparison

First of all, in order to test the feature integration power of our method, we compare classification performance using all the feature modalities with that using only one feature modality. From Table 4.4 to Table 4.6, we can draw the conclusion that the performance of our proposed AMMSS can beat the best of single modality, which tackles the problem of Eq. (4.22).

We also compare our methods with some graph based state-of-the-art semi-supervised learning methods: (a) the harmonic function (HF) approach [79], (b) learning with local and global consistency approach (LGC) [80] and (c) the random walk approach (RW) [81]. For each of the above three methods, we use the kernel addition (KA), that is, the simple average of equal weighted Laplacian matrices or the graph Laplacian of the concatenated features of all modalities (FC) as the input for HF, LGC as well as RW. Moreover, for sake of completeness, we also compare the results of support vector machine with the pre-computed kernel Eq. (4.20) implemented by LIBSVM [59]. Since Multiple Kernel Learning (MKL) approaches [82] can also realize feature integration if we consider one feature modality as one kernel, we report its classification result as well. Moreover, since our method can learn the weight for each feature modality adaptively, we compare the results of our model using equal weight (MMSS). We adopt the optimal parameter settings for the above methods empirically. As for performance evaluation, we utilize the widely-used performance metrics, average macro classification accuracy as well as average micro classification accuracy for each class. Average macro classification accuracy is shown in Table 4.7 and micro accuracy for all the datasets are shown in Fig. 4.7. We can see that our method always achieves consistently better results than the other state-of-art methods in



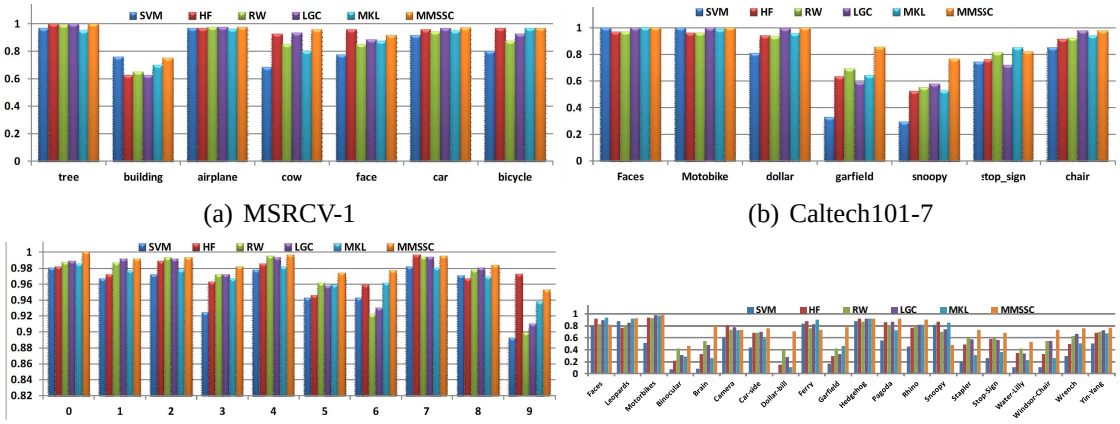
(a) avg. accuracy **94**% (b) avg. accuracy **91**% (c) avg. accuracy **98**%

Figure 4.6. Calculated confusion matrices by AMMSS method (a) MSRCV1 (b) Caltech101-7 (c) Handwritten numerals..

terms of average macro classification accuracy and choosing different weights for different features can even boost the performance of multi modality semi-supervised learning results. As for average micro classification accuracy, the results of AMMSS are the best for most classes. The confusion matrices of MSRCV1, Caltech101-7 and Handwritten numerals are shown in Fig. 4.6.

Moreover, since our method can learn the weight for each feature modality after convergence, we add the generalization ability of the objective function Eq. (4.25). Fig. 4.8 shows the learned weight by our Alg. 7 on five benchmark datasets. From it, we can observe that DoG-SIFT has the most discriminate power in Caltech$101-7$ dataset, CENTRIST has the highest weight for Caltech$101-20$ dataset while for MSRCV1 dataset, GIST is the best feature modality among the six which is consistent with single modality's performance shown in Table 4.4. Instead of treating each feature modality equally, our method can do weighting each feature modality and classification simultaneously.

At last, we test the convergency speed of our AMMSS algorithm, which is shown in Fig. 4.9. From it, we can observe that our AMMSS algorithm converges very fast on all the datasets and usually the number of iteration is less than 10.



(a) MSRCV-1 (b) Caltech101-7

(c) handwritten number (d) Caltech101-20

Figure 4.7. The Micro accuracy on two datasets (a) MSRCV1 (b) Caltech101-7. (c) Handwritten number (d) Caltech101-20.

Table 4.4. The average macro classification accuracy compared with single view on Caltech101-7, Caltech101-20 and MSRCV1 datasets.

Methods	Caltech7	Caltech20	MSRCV1
CTM [49]	0.45	0.27	0.30
LBP [46]	0.66	0.39	0.71
GIST [47]	0.80	0.51	0.79
CENTRIST [48]	0.79	0.70	0.77
DoG-SIFT [44]	0.81	0.30	0.51
HOG [45]	0.89	0.27	0.69
AMMSS	**0.91**	**0.74**	**0.94**

Table 4.5. The average macro classification accuracy compared with single view on Handwritten numerals dataset.

Data	FOU	FAC	KAR	PIX	ZER	MOR	AMMSS
HW	0.92	0.82	0.93	0.46	0.94	0.82	**0.98**

Table 4.6. The average macro classification accuracy compared with single view on animal with attribute dataset.

Data	CQ	LSS	PHOG	RGISIFT	SIFT	SURF	AMMSS
AWA	0.057	0.062	0.050	0.054	0.065	0.072	**0.095**

Table 4.7. The average macro classification accuracy compared with baseline methods on all datasets.

Methods	Caltech7	Caltech20	MSRCV1	HW	AWA
SVM [59]	0.85	0.59	0.86	0.95	0.076
MKL [82]	0.89	0.68	0.89	0.96	0.079
HF(KA) [79]	0.84	0.70	0.92	0.97	0.079
HF(FC) [79]	0.82	0.68	0.89	0.96	0.077
RW(KA) [81]	0.89	0.72	0.88	0.97	0.080
RW(FC) [81]	0.86	0.69	0.87	0.96	0.079
LGC(KA) [83]	0.87	0.72	0.90	0.97	0.081
LGC(FC) [83]	0.89	0.71	0.88	0.96	0.079
MMSS	0.89	0.72	0.92	0.97	0.086
AMMSS	**0.91**	**0.74**	**0.94**	**0.98**	**0.095**

4.4 Conclusion

In this chapter, we proposed two graph based methods to fuse heterogeneous image features. One is to do unsupervised spectral clustering and the other is to do semi-supervised learning. Utilizing our algorithms, a common cluser/class indicator matrix will be learned. And by decomposing the original problem into several subproblems, we can solve the proposed model iteratively with the proof of convergence to local/global solution. Empirical studies have been conducted on bench-mark data sets. Compare with the existing state-of-art methods, our proposed models consistently achieve better clustering or classification performance.

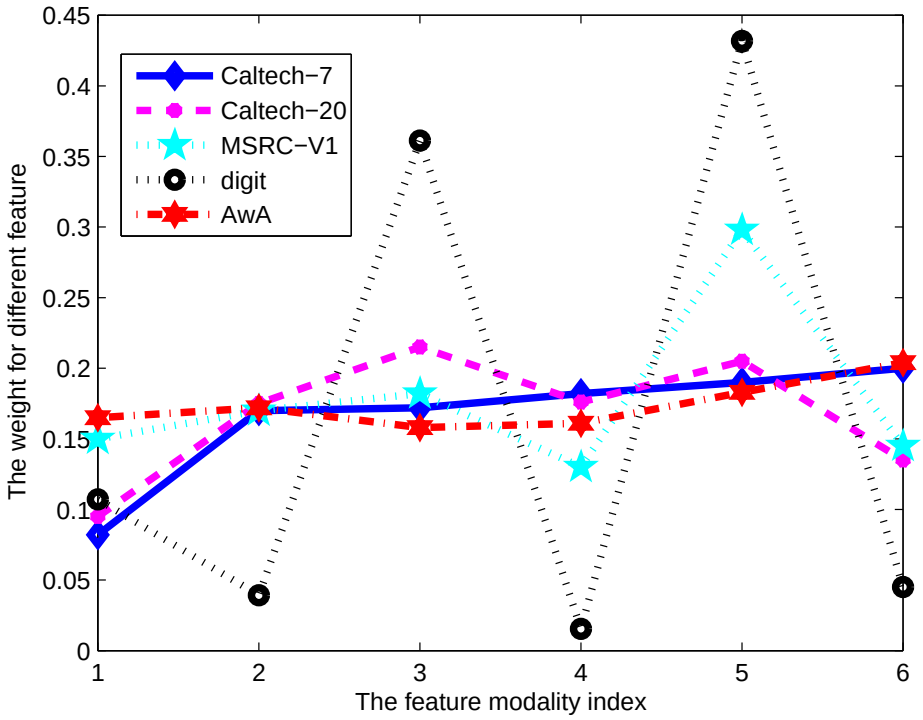
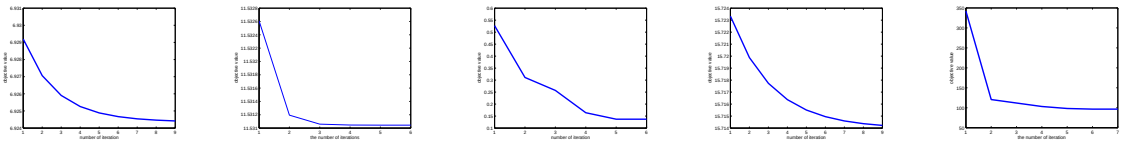


Figure 4.8. The learned weight factor for different modalities on five dataset. The feature index on x-axis from 1 to 6 stands for CMT, LBP, GIST, HOG, CENTRIST and DOG-SIFT respectively for Caltech-7, Caltech-20 and MSRCV1 datasets. And the index on x-axis from 1 to 6 stands for FOU, FAC, KAR, PIX, ZER, MOR respectively for Handwritten numerals dataset. The index on x-axis from 1 to 6 stands for CQ, LSS, PHOG, RGSIFT, SIFT, SURF respectively for AwA dataset..



(a) Caltech101−7 (b) Caltech101 − 20 (c) MSRCV1 (d) Handwritten (e) AwA

Figure 4.9. The convergency of five datasets (a) Caltech101-7 (b) Caltech101-20 (c) MSRCV1 (d) Handwritten numerals (e) AwA.

Algorithm 7 The algorithm of AMMSS

Input:

1. Affinity matrices $\{W^{(1)}, \cdots, W^{(V)}\} \in \mathbb{R}^{n \times n}$

2. The labels for the first l images, $Y_l = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_l]^T$, $\mathbf{y}_i \in \mathbb{B}^{K \times 1}, \quad \forall i = 1, 2, ..., l$.

3. The parameters r and λ.

Output:

1. The predicted labels for the unlabeled images $y_i, \quad \forall i = l+1, l+2, ..., n$.

2. The weight scalar $\alpha^{(v)}, \quad \forall v = 1, 2, ..., V$ for each modality.

Initialization:

1. Set $t = 0$

2. Initialize the weight for each modality, $\alpha_t^{(v)} = \frac{1}{V}, \quad \forall v = 1, 2, ..., V$

3. Initialize the common class label matrix, $G_t = \begin{bmatrix} G_{l_t} \\ G_{u_t} \end{bmatrix} = \begin{bmatrix} Y_l \\ Y_u \end{bmatrix}$ where $Y_u \in \mathbb{R}^{u \times K}$ is a random matrix and each entry $u_{i,j} \in \{0, 1\}$.

4. Calculate the normalized Laplacian matrices for each feature modality, $L_t^{(v)} = I - (D_t^{(v)})^{-\frac{1}{2}} W_t^{(v)} (D_t^{(v)})^{-\frac{1}{2}}$

Procedure:

repeat

1. Calculate $\tilde{L}_t^{(v)} = (\alpha_t^{(v)})^r L_t^{(v)}$

2. Calculate the class indicator matrix for each modality $G_t^{(v)} = \lambda(\tilde{L}_t^{(v)} + \lambda I)^{-1} G_t$

3. Calculate $H_t = \sum_{v=1}^{V} \left(I - \lambda(\tilde{L}_t^{(v)} + \lambda I)^{-1} \right)$ and split the H_t by Eq. (4.35).

4. Calculate $p_t^{(v)} = \text{Tr}\ ({G_t^{(v)}}^T L_t^{(v)} G_t^{(v)})$

5. Update the weight for each modality by Eq. (4.30)

6. Update $G_{u_{t+1}} = -H_{uu_t}^{-1} H_{ul_t} Y_l$. And update $G_{t+1} = \begin{bmatrix} Y_l \\ G_{u_t} \end{bmatrix}$

7. Update $t = t + 1$

until Converges

Assign the single class label for the unlabeled images by Eq. (4.39).

CHAPTER 5

ON THE EQUIVALENT OF LOW-RANK LINEAR REGRESSIONS AND LINEAR DISCRIMINANT ANALYSIS BASED REGRESSIONS

5.1 Introduction

As one of most important data mining and machine learning technique, multivariate linear regression attempts to model the relationship between predictors and responses by fitting a linear equation to observed data. Such linear regression models suffer from two deficiencies when they are applied to the real-world applications. First, the linear regression models usually have low performance for analyzing the high-dimensional data. In many data mining and machine learning applications, such as gene expression, document classification, face recognition, the input data have a large number of features. To perform accurate regression or classification tasks on such data, we have to collect an enormous number of samples. However, due to the data and label collection difficulty, we often cannot obtain enough samples and suffer from the curse-of-dimensionality problem [84]. To solve this problem, the dimensionality reduction methods, such as linear discriminant analysis (LDA) [85], were often used to reduce the feature dimensionality first.

Second, the linear regression models don't emphasize the correlations among different responses. Standard least squares regression is equivalent to regressing each response on the predictors separately. To incorporate the response (*i.e.* classes or tasks) correlations into the regression model, Anderson introduced the reduced rank regression method [86], which is a multivariate regression model with a coefficient matrix with reduced rank. Later many researchers worked on the low-rank (or reduced) regression models [86–91], in which

the classes/tasks correlation patterns are explored by the low-rank structure and utilized to enhance the regression/classification results.

In this chapter, we propose new and important theoretical foundations of the low-rank regression. We first present the discriminant low-rank linear regression, which reformulates the standard low-rank regression to a more interpretable objective. After that, we prove that the low-rank regression model is indeed equivalent to doing linear regression in the LDA subspace, *i.e.* the learned low-rank classes/tasks correlation patterns are connected to the LDA projection results. Our new theorem explains the underlying computational mechanism of low-rank regression, which performs the LDA projection and the linear regression on data points simultaneously. In our special case, when the low-rank regression coefficient matrix becomes a full-rank matrix, our result is connected to Ye's work on the equivalence between the multivariate linear regression and LDA [92].

Motivated by our new theoretical analysis, we propose two new discriminant low-rank regression models, including low-rank ridge regression (LRRR) and sparse low-rank regression (SLRR). Both methods are equivalent to performing the regularized regression tasks in the regularized LDA subspace (two methods have different regularization terms). Because the regularization term avoids the rank deficiency problem in both regression and LDA, our LRRR method outperforms the low-rank regression in both theoretical analysis and experimental results. Using the structured sparsity-inducing norm based regularization term, our SLRR method can explore both classes/tasks correlations and feature structures. All our new discriminant low-rank regression models can simultaneously analyze the high-dimensional data in the discriminant subspace without any pre-processing step and incorporate the classes/tasks correlations. We evaluate the proposed methods on six benchmark data sets. In all experimental results, our discriminant low-rank models consistently outperform their corresponding full-rank counterparts.

5.2 Low-Rank Regression and LDA+LR

One of the main result of this paper is to prove that the low-rank linear regression (LRLR) is equivalent to doing standard linear regression in LDA subspace (we call this as "LDA+LR").

5.2.1 Low-Rank Linear Regression Revisit

Traditional Linear Regression model for classification is to solve the following problem:

$$\min_{W} ||Y - X^T W||_F^2 , \tag{5.1}$$

where $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \Re^{d \times n}$ is the centered training data matrix and $Y \in \Re^{n \times k}$ is the normalized class indicator matrix, i.e. $Y_{i,j} = 1/\sqrt{n_j}$ if the i-th data point belongs to the j-th class and $Y_{i,j} = 0$ otherwise and n_j is the sample size of the j-th class. The model outputs the parameter matrix $W \in \Re^{d \times k}$, which can be used to predict any test data point $\mathbf{x} \in \Re^{d \times 1}$ by $W^T\mathbf{x}$.

When the class or task number is large, there are often underlying correlation structures between classes or tasks. To explore these hidden structures and utilize such patterns to improve the learning model, in recent work [38], researchers presented to learn a low-rank projection W in the regression model by imposing the trace norm regularization as:

$$\min_{W} ||Y - X^T W||_F^2 + \lambda ||W||_* . \tag{5.2}$$

The trace norm regularization can discover the low-rank structures existing between classes or tasks. Using Eq. (5.2), the rank of coefficient matrix W, which is decided by the selection of parameter λ, cannot be explicitly selected and tuned.

In related research work, the low-rank regression was studied in statistics and machine learning communities [86–91]. In the low-rank regression, the rank of W is explicitly

decided by constraining the rank of W to be $s < min(n, k)$ and solving the following problem:

$$\min_{W} ||Y - X^T W||_F^2, \;\; s.t. \;\; rank(W) \leq s. \tag{5.3}$$

Because the rank of coefficient matrix can be explicitly determined, the low-rank regression in Eq. (5.3) is better than the trace norm based objective in Eq. (5.2) in practical applications. Although the general rank minimization is a non-convex and NP-hard problem, the objectives with rank constraints are solvable, *e.g.* the global solution was given in [87, 88].

5.2.2 Relation to LDA+LR

In this section, we will show that the low-rank linear regression (LRLR) is equivalent to perform Linear Discriminant Analysis (LDA) and linear regression simultaneously (LDA+LR). In other words, the learned low-rank structures and patterns are induced by the LDA projection (with regression). The low rank s is indeed the projection dimension of LDA.

Before introducing our main theorems, we first propose the following discriminant Low-Rank Linear Regression formulation (LRLR):

$$\min_{A,B} ||Y - X^T AB||_F^2 , \tag{5.4}$$

where $A \in \Re^{d \times s}$, $B \in \Re^{s \times k}$, $s < min(d, k)$. Thus $W = AB$ has low-rank s. The above LRLR objective has the same solutions as Eq. (5.3), but it has clearer discriminant projection interpretation. Eq. (5.4) can be written as

$$\min_{A,B} ||Y - (A^T X)^T B||_F^2 . \tag{5.5}$$

This shows A can be viewed as a projection. Interestingly as we show in Theorem 1, A is exactly the optimal subspace defined by the classic LDA.

Theorem 2 *The low-rank linear regression method of Eq. (5.4)) is identical to doing standard linear regression in LDA subspace.*

Proof: Denoting $J_1(A, B) = ||Y - X^T AB||_F^2$ and taking its derivative w.r.t. B, we have,

$$\frac{\partial J_1(A, B)}{\partial B} = -2A^T XY + 2A^T XX^T AB. \tag{5.6}$$

Setting Eq. (5.6) to zero, we obtain,

$$B = (A^T XX^T A)^{-1} A^T XY. \tag{5.7}$$

Substituting Eq. (5.7) back into Eq. (5.4), we have,

$$\min_A ||Y - X^T A(A^T XX^T A)^{-1} A^T XY||_F^2, \tag{5.8}$$

which is equivalent to

$$\max_A \text{Tr}\ ((A^T (XX^T)A)^{-1} A^T XYY^T X^T A). \tag{5.9}$$

Note that

$$S_t = XX^T, \quad S_b = XYY^T X^T, \tag{5.10}$$

where S_t and S_b are the total-class scatter matrix and the between-class scatter matrix defined in the LDA, respectively. Therefore, the solution of Eq. (5.9) can be written as:

$$A^* = \arg\max_A\ \text{Tr}\ [(A^T S_t A)^{-1} A^T S_b A], \tag{5.11}$$

which is exactly the problem of LDA, and the global optimal solution to Eq. (5.11) is the top s eigenvectors of $S_t^{-1} S_b$ corresponding to the nonzero eigenvalues (if S_t is singular, we compute the eigenvectors of $S_t^+ S_b$ corresponding to the nonzero eigenvalues, where S_t^+ denotes the pseudo-inverse of S_t). Now Eq. (5.5) implies that we do linear regression on the projected data $\widetilde{X} = A^T B$. Since A is the LDA projection, thus Eq. (5.5) implies we do regression on the LDA subspace.

□

Note that in Eq. (5.4), the class indicator matrix Y is normalized, but not centered. However X is centered. The following Theorem 3 shows that we obtain the optimal solution whatever Y is centered or not.

Theorem 3 *The optimal solution* (A^*, B^*) *for the following problem*

$$\min_{A,B} ||PY - X^T AB||_F^2 \tag{5.12}$$

is identical to those of Eq. (5.4); here $P = I - \boldsymbol{e}\boldsymbol{e}^T/n \in \Re^{n\times n}$ *is the centering matrix, and* $\boldsymbol{e} = (1 \cdots 1)^T$.

For this reason, the bias (intercept) term are already automatically incorporated in Eq. (5.4).

Proof: The key point of the proof is the fact that in the solution for both B and A of Eq. (5.7) and Eq. (5.9), Y always appears together with X as combination

$$XY = (XP)Y = XP^2Y = (XP)(PY),$$

because X is centered and $P^2 = P$. In other words, as long as X is centered, Y is automatically centered. □

This results can be easily extended to the standard linear regression. In fact we have

Remark **1** *As long as* X *is centered, the optimal solution* W^* *for the standard linear regression of Eq.(1) remains identical no matter* Y *is centered or not.*

Our new results provide the theoretical foundation to explain the mechanism behind the low-rank regression methods. Meanwhile, the above proof process also indicates a concise algorithm to achieve the global solution of LRLR in Eq. (5.4), as well as Eq. (5.3). The Algorithm to solve Eq. (5.4) is summarized in Alg. 8.

Moreover, we note that Theorem 2 also provides clarification to a long-standing puzzle in multi-class LDA, as explained below.

Algorithm 8 The algorithm to solve LRLR

Input:

1. The centralized training data $X \in \Re^{d \times n}$.

2. The normalized training indicator matrix $Y \in \Re^{n \times k}$.

3. The low-rank parameter s.

Output:

1. The matrices $A \in \Re^{d \times s}$ and $B \in \Re^{s \times k}$.

Process:

1. Calculate A by Eq. (5.11)

2. Calculate B by Eq. (5.7)

5.2.3 LDA: Trace-of-Ratio or Ratio-of-Trace?

The original Fisher LDA is on 2-class problem, where only $k - 1 = 1$ projection direction **a** is needed. The Fisher objective is

$$\max_{\mathbf{a}} \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_w \mathbf{a}}.$$

The generalization to multi-class has two natural formulations [85], either the trace-of-ratio formulation

$$\max_{A} \text{Tr} \frac{A^T S_b A}{A^T S_w A} \tag{5.13}$$

where $A = (\mathbf{a}_1 \cdots \mathbf{a}_{k-1})$, or the ratio-of-trace formulation[1]

$$\max_{A} \frac{\text{Tr } A^T S_b A}{\text{Tr } A^T S_w A} \tag{5.14}$$

Our Theorem 2 lends support to the trace-of-ratio objective function because this formulation arises directly from the linear regression.

[1] In Eqs.(5.13,5.14), the optimal solution remains the same when S_w is replaced by S_t.

5.2.4 Full-Rank Linear Regression and LDA

Here we note an important connection. In the special case, the low-rank regression coefficient matrix W becomes a full-rank matrix. Without loss of generality we assume $s = k \leq n$, because the number of data points n is usually larger than the number of classes k. The matrix $B \in \Re^{k \times k}$ becomes a square matrix. Because $rank(W) = rank(AB) = k$ and $k \leq n$, $rank(A) \geq k$ and $rank(B) \geq k$. Thus, $rank(B) = k$ and B is a full rank matrix, *i.e.* the matrix B is invertible.

The Theorem 2 is still correct for the special case. Moreover, we can further conclude the equivalence between the multivariate linear regression and LDA results. We can simply prove this conclusion. Because the matrix A includes the LDA subspaces and the matrix B can be considered as an invertible rotational matrix, thus AB is also one of the infinite number global solutions of LDA [93]. Thus, in the special full-rank case, the multivariate linear regression is equivalent to the LDA result, which was shown in Ye's work [92] with the assumptions: the reduced dimension is $k - 1$ and $rank(S_b) + rank(S_w) = rank(S_t)$. Our proof is more general and doesn't need the rank assumption.

5.2.5 Low-Rank Ridge Regression (LRRR)

As we know, by adding a Frobenius norm based regularization on the linear regression loss, ridge regression can achieve better performance than linear regression [94]. Thus, it is important and necessary to add the ridge regularization into low-rank regression formulation. We propose the following Low-Rank Ridge Regression (LRRR) objective as,

$$\min_{A,B} ||Y - X^T AB||_F^2 + \lambda ||AB||_F^2 , \tag{5.15}$$

where $A \in \Re^{d \times s}$, $B \in \Re^{s \times k}$, $s < min(n, k)$, λ is the regularization parameter. Similarly, we can see that the LRRR objective is equivalent to the following objective:

$$\min_{W} ||Y - X^T W||_F^2 + \lambda ||W||_F^2, \quad s.t. \quad rank(W) \leq s. \tag{5.16}$$

Compared to Eq. (5.16), Eq. (5.15) provides better chance for us to understand the learning mechanism of LRRR. We will show that our new LRRR objective is connected to the regularized discriminant analysis, which provides better projection results than the standard LDA. We will also derive the global solution of the non-convex problems in Eq. (5.15) and Eq. (5.16).

Theorem 4 *The proposed Low-Rank Ridge Regression (LRRR) method (both Eq. (5.15) and Eq. (5.16)) is equivalent to doing the regularized regression in the regularized LDA subspace.*

Proof: Denoting $J_2(A,B) = ||Y - X^TAB||_F^2 + \lambda||AB||_F^2$, and taking its derivative w.r.t. B, we have,

$$\frac{\partial J_2(A,B)}{\partial B} = -2A^TXY + 2A^TXX^TAB + 2\lambda A^TAB. \tag{5.17}$$

Setting Eq. (5.17) to zero, we get,

$$B = (A^T(XX^T + \lambda I)A)^{-1}A^TXY, \tag{5.18}$$

where $I \in \Re^{d\times d}$ is the identity matrix. Substituting Eq. (5.18) back into Eq. (5.15), we have

$$\begin{aligned}\min_A ||Y - X^TA(A^TXX^TA + \lambda A^TA)^{-1}A^TXY||_F^2 \\ +\lambda||A(A^T(XX^T + \lambda I)A)^{-1}A^TXY||_F^2,\end{aligned} \tag{5.19}$$

which is equivalent to the following problem:

$$\max_A\{(A^T(XX^T + \lambda I)A)^{-1}A^TXYY^TX^TA\}. \tag{5.20}$$

Similarly, the solution of Eq. (5.20) can be written as:

$$A^* = \arg\max_A\{Tr((A^T(S_t + \lambda I)A)^{-1}A^TS_bA)\}, \tag{5.21}$$

Algorithm 9 The algorithm to LRRR

Input:

1. The centralized training data $X \in \Re^{d \times n}$.

2. The normalized training indicator matrix $Y \in \Re^{n \times k}$.

3. The low-rank parameter s.

4. The regularization parameter λ.

Output:

1. The matrices $A \in \Re^{d \times s}$ and $B \in \Re^{s \times k}$.

Process:

1. Calculate A by Eq. (5.21)

2. Calculate B by Eq. (5.18)

which is exactly the problem in regularized LDA [95]. After we get the optimal solution A, we can re-write Eq. (5.15) as:

$$\min_B ||Y - (A^T X)^T B||_F^2 + \lambda ||AB||_F^2, \tag{5.22}$$

which is the regularized regression, and the optimal solution is given by Eq. (5.18). Thus, the LRRR of Eq. (5.15) is equivalent to performing ridge regression in regularized-LDA subspace. □

Similar to Theorem 3, we can show that Y is automatically centered as long as X is centered.

Another interest point is that although our LRRR model is a non-convex problem, Theorems 1 and 3 show that they have the global optimal solutions. The Algorithm to solve LRRR of Eq. (5.15) is described in Alg. 9.

5.2.6 Full-Rank Ridge Regression and Regularized LDA

In the special case, the low-rank regression coefficient matrix W becomes a full-rank matrix. Similar to §5.2.4, we have the following lemma:

Lemma 1 *The full-rank ridge regression result is equivalent to the solution of regularized LDA (S_t is replaced by the regularized form $S_t + \lambda I$).*

Similar to the proof in §5.2.4, we can easily prove the coefficient matrix W in full-rank ridge regression is one of the global solutions of LDA regularized by λI.

5.3 Sparse Low-Rank Regression for Feature Selection

Besides exploring and utilizing the class/task correlations and structure information, the learning models also prefer to select and use the important features to avoid the "curse of dimensionality" problem in high-dimensional data analysis. Thus, it is important to extend our discriminant low-rank regression formulations to feature selection models.

Due to the intrinsic properties of real world data, the structured sparse learning models have shown superior feature selection results in previous research [19, 30, 33, 96–101]. One of the most effective ways for selecting features is to impose sparsity by inducing hybrid structured $\ell_{2,1}$-norm on the coefficient matrix W as the regularization term [16, 38]. Therefore, following our LRLR and LRRR methods, we propose a new Sparse Low-Rank Regression (SLRR) method, which reserves the low-rank constraint and adds the mixed $\ell_{2,1}$-norm regularization term to induce both desired low-rank structure of classes/tasks correlations and structured sparsity between features. To be specific, "low-rank" means $rank(AB) = s < min(n, k)$ and "structured sparsity" means most rows of AB are zero to help feature selection. Thus, we solve:

$$\min_{A,B} ||Y - X^T AB||_F^2 + \lambda ||AB||_{2,1} \,, \tag{5.23}$$

where $A \in \Re^{d\times s}$, $B \in \Re^{s\times k}$, $s < min(n, k)$. Similarly, we can see that the above SLRR objective is equivalent to the following objective:

$$\min_{W} ||Y - X^T W||_F^2 + \lambda||W||_{2,1}, \quad s.t. \quad rank(W) \leq s. \tag{5.24}$$

Both Eq. (5.23) and Eq. (5.24) are new objectives to simultaneously learn low-rank classes correlation patterns and features structured sparsity.

5.3.1 Connection to Discriminant Analysis

Interestingly our new SLRR method also connects to the regularized discriminant analysis by the following theorem.

Theorem 5 *The optimal solution of the proposed SLRR method (Eq. (5.23) and Eq. (5.24)) has the same column space of a special regularized LDA.*

Proof: Eq. (5.23) is equivalent to the following problem,

$$\min_{A,B} ||Y - X^T AB||_F^2 + \lambda \text{Tr}\,(B^T A^T DAB), \tag{5.25}$$

where $D \in \Re^{d\times d}$ is a diagonal matrix and each element on the diagonal is defined as follows:

$$d_{ii} = \frac{1}{2||\mathbf{g}^i||_2}, \quad i = 1, 2, ..., d, \tag{5.26}$$

where $\mathbf{g}^i$ is the i-th row of matrix $G = A^*B^*$. Denoting $J_3(A, B) = ||Y - X^T AB||_F^2 + \lambda \text{Tr}\,(B^T A^T DAB)$ and taking its derivative w.r.t. B, we have,

$$\frac{\partial J_3(A, B)}{\partial B} = -2A^T XY + 2A^T XX^T AB + 2\lambda A^T DAB. \tag{5.27}$$

Setting the above equation to be zero, we can get,

$$B = (A^T(XX^T + \lambda D)A)^{-1} A^T XY, \tag{5.28}$$

where $D \in \Re^{d \times d}$ is the diagonal matrix defined in Eq. (5.26). Substituting Eq. (5.28) back into Eq. (5.25), then we need solve the following problem to get A,

$$\max_A \text{Tr}\left((A^T(XX^T + \lambda D)A)^{-1}A^T XYY^T X^T A\right). \tag{5.29}$$

The solution of Eq. (5.29) is:

$$A^* = \arg\max_A \{\text{Tr}\left((A^T(S_t + \lambda D)A)^{-1}A^T S_b A\right)\}, \tag{5.30}$$

Since the column space of $W^* = A^*B^*$ is identical to the column space of A^*, the proposed SLRR has the same column space of a special regularized LDA (S_t is replaced with $S_t + \lambda D$). □

After we get the optimal solution A, we can solve Eq. (5.23) through Eq. (5.25), which is the regularized regression problem. Again, similar to Theorm 3, we can prove that if Y is centered or not will not affect the learnt model A^* and B^*.

5.3.2 Algorithm to Solve SLRR

Solving SLRR objective in Eq. (5.23) is nontrivial, there are two variables A and B needed to be optimized, and the non-smooth regularization also makes the problem more difficult to solve. Interestingly, a concise algorithm can be derived to solve this problem based on the above proof. The detailed algorithm is described in Algorithm 10. In next subsection, we will prove that the algorithm converges. Our experimental results show that the algorithm always converges in 5-20 iterations.

5.3.3 Algorithm Convergence Analysis

Because Alg. 10 is an iterative algorithm, we will prove its convergence.

Theorem 6 *Alg. 10 decreases the objective function of Eq. (5.23) monotonically.*

Algorithm 10 The algorithm to SLRR

Input:

1. The centralized training data $X \in \Re^{d \times n}$.

2. The normalized training indicator matrix $Y \in \Re^{n \times k}$.

3. The low-rank parameter s.

4. The regularization parameter λ.

Output:

1. The matrices $A \in \Re^{d \times s}$ and $B \in \Re^{s \times k}$.

Initialization:

1. Set $t = 0$

2. Initialize $D^{(t)} = I \in \Re^{d \times d}$.

Repeat:

1. Calculate $A^{(t+1)}$ by Eq. (5.30)

2. Calculate $B^{(t+1)}$ by Eq. (5.28)

3. Update the diagonal matrix $D^{(t+1)} \in \Re^{d \times d}$, where the i-th diagonal element is $\frac{1}{2||(A^{(t+1)}B^{(t+1)})^i||_2}$.

4. Update $t = t + 1$

Until Converge.

Proof: In the t-th iteration, we have

$$\begin{aligned} &< A^{(t+1)}, B^{(t+1)} >= \arg\min_{A,B} ||Y - X^T AB||_F^2 \\ &+\lambda \text{Tr}\,(B^T A^T D^{(t)} AB) \end{aligned} \tag{5.31}$$

In other words,

$$\begin{aligned} &||Y - X^T A^{(t+1)} B^{(t+1)}||_F^2 + \lambda \text{Tr}\,(B^{(t+1)^T} A^{(t+1)^T} D^{(t)} A^{(t+1)} B^{(t+1)}) \\ &\leq ||Y - X A^{(t)} B^{(t)}||_F^2 + \lambda \text{Tr}\,(B^{(t)^T} A^{(t)^T} D^{(t)} A^{(t)} B^{(t)}) \end{aligned} \tag{5.32}$$

Denote $G^{(t)} = A^{(t)}B^{(t)}$ and $G^{(t+1)} = A^{(t+1)}B^{(t+1)}$. By the definition of matrix D in the algorithm, Eq. (5.32) can be rewritten as,

$$\begin{aligned} &||Y - X^T G^{(t+1)}||_F^2 + \lambda \sum_{i=1}^{d} \frac{||\mathbf{g}^{i(t+1)}||_2^2}{2||\mathbf{g}^{i(t)}||_2} \\ &\leq ||Y - X^T G^{(t)}||_F^2 + \lambda \sum_{i=1}^{d} \frac{||\mathbf{g}^{i(t)}||_2^2}{2||\mathbf{g}^{i(t)}||_2} \end{aligned} \tag{5.33}$$

where $\mathbf{g}^{i(t)}$ and $\mathbf{g}^{i(t+1)}$ are the i-th row of the matrix $G^{(t)}$ and $G^{(t+1)}$ respectively. Since for each i, we have

$$||\mathbf{g}^{i(t+1)}||_2 - \frac{||\mathbf{g}^{i(t+1)}||_2^2}{2||\mathbf{g}^{i(t)}||_2} \leq ||\mathbf{g}^{i(t)}||_2 - \frac{||\mathbf{g}^{i(t)}||_2^2}{2||\mathbf{g}^{i(t)}||_2}. \tag{5.34}$$

Thus, summing up d inequalities and multiplying the summation with the regularization parameter λ, we obtain:

$$\begin{aligned} &\lambda \sum_{i=1}^{d} \left(||\mathbf{g}^{i(t+1)}||_2 - \frac{||\mathbf{g}^{i(t+1)}||_2^2}{2||\mathbf{g}^{i(t)}||_2} \right) \\ &\leq \lambda \sum_{i=1}^{d} \left(||\mathbf{g}^{i(t)}||_2 - \frac{||\mathbf{g}^{i(t)}||_2^2}{2||\mathbf{g}^{i(t)}||_2} \right) \end{aligned} \tag{5.35}$$

Combining Eq. (5.33) and Eq. (5.35), we get:

$$\begin{aligned} &||Y - X^T G^{(t+1)}||_F^2 + \lambda \sum_{i=1}^{d} ||\mathbf{g}^{i(t+1)}||_2 \\ &\leq ||Y - X^T G^{(t)}||_F^2 + \lambda \sum_{i=1}^{d} ||\mathbf{g}^{i(t)}||_2 \end{aligned} \tag{5.36}$$

Therefore, we have:

$$||Y - X^T G^{(t+1)}||_F^2 + \lambda ||G^{(t+1)}||_{2,1} \leq ||Y - XG^{(t)}||_F^2 + \lambda ||G^{(t)}||_{2,1} \tag{5.37}$$

Since A and B are updated according to gradient, Alg. 10 will monotonically decrease the objective in Eq. (5.23) in each iteration. □

5.3.4 Full-Rank Sparse Linear Regression and Regularized LDA

In the special case, the low-rank regression coefficient matrix W becomes a full-rank matrix. Similar to §5.2.4, we also have the following lemma:

Lemma 2 *The optimal solution of the full-rank sparse linear regression is one of the global solutions of LDA regularized by λD.*

Similar to the proof in §5.2.4, we can easily prove the coefficient matrix W in full-rank sparse linear regression is one of the global solutions of LDA regularized by λD.

5.4 Experimental Results

In this section, we will evaluate the performance of our proposed LRLR, LRRR, SLRR with their corresponding full-rank counterparts. We firstly introduce the six benchmark datasets used in our experiments.

5.4.1 Dataset Descriptions

UMIST face dataset [102] contains 20 persons and totally 575 images. All images are cropped and resized into 112×92 pixels per image.

Binary Alphadigits 36 dataset [103] contains binary digits of 0 through 9 and capital A through Z with size 20×16. There are 39 examples of each class.

Binary Alphadigits 26 dataset [103] contains binary capital A through Z with size 20×16. There are 39 examples of each class.

VOWEL dataset [104] consists of 990 vowel recognition data used for the study of recognition of the eleven steady state vowels of British English. The speakers are indexed by integers 0-89. (Actually, there are fifteen individual speakers, each saying each vowel six times.) The vowels are indexed by integers 0-10. For each utterance, there are ten floating-point input values, with array indices 0-9.

MNIST hand-written digits dataset [105] consists of $60,000$ training and $10,000$ testing digits. It has 10 classes, from digit 0 to 9. Each image is centralized (according to the center of mass of the pixel intensities) on a 28×28 grid. We randomly select 15 images for each class in our experiment.

JApanese Female Facial Expressions (JAFFE) data set [106] contains 213 photos of 10 Japanese female models. Each image has been rated on 6 emotion adjectives by 60 Japanese subjects.

We summarize the datasets that we will use in our experiments in Table 5.1

5.4.2 Experimental Setup

All the datasets in our experiments have large number of classes (at least 10 classes). For each dataset, we randomly split the data into 5 parts. According to the standard 5-fold cross validation, in each round, we use 4 parts for training and the remaining part for testing. The average classification accuracy rates for different methods are reported. In the training stage, we use different full-rank linear regression models, *i.e.* full-rank linear regression, full-rank ridge regression, sparse full-rank linear regression to learn the coefficient matrix W directly or we solve the proposed low-rank counterparts (LRLR, LRRR, SLRR) to calculate W indirectly by $W = AB$. In all experiments, we automatically tune the regularization parameters by selecting the best parameters among the values $\{10^r : r \in \{-5, -4, -3, ...3, 4, 5\}\}$ with 5-fold cross validation on the corresponding training data only. In addition, for LRLR, LRRR, SLRR, we calculate the classification results with respect to different low-rank parameters s in the range of $[k/2, k)$, where k is the number of classes. At last, in the testing stage, we utilize the following decision function to classify the coming testing data $\mathbf{x}_t \in \Re^{d\times 1}$ into one and only one out of k classes,

$$\arg\max_{1\leq j\leq k} \left(W^T \mathbf{x}_t\right)_j . \tag{5.38}$$

Table 5.1. The summary of the datasets used in our experiments. k is the number of classes, d is the number of feature dimensions, n is the number of data points.

Dataset	k	d	n
UMIST	20	10304	575
BINALPHA36	36	320	1404
BINALPHA26	26	320	1014
VOWEL	11	10	990
MNIST	10	784	150
JAFFE	10	1024	213

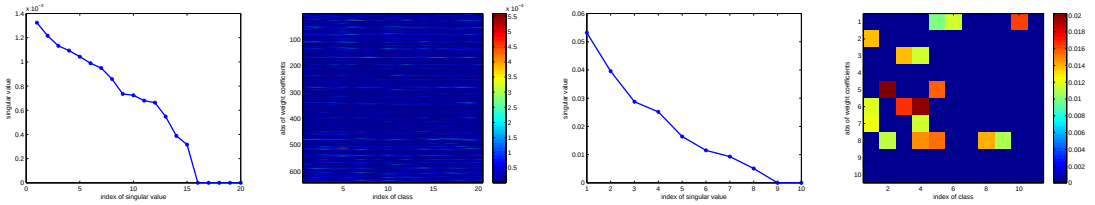
Table 5.2. The average classification accuracy using different regression methods on six datasets.

Data	Rank	Linear Regression(LR)	Ridge Regression(RR)	Sparse Regression(SR)
UMIST	Full	0.6650 ± 0.1069	0.9197 ± 0.0456	0.9525 ± 0.0533
	Low	$\mathbf{0.8225 \pm 0.0937}$	$\mathbf{0.9675 \pm 0.0322}$	$\mathbf{0.9675 \pm 0.0245}$
BINALPHA36	Full	0.3488 ± 0.0241	0.6039 ± 0.0231	0.5971 ± 0.0205
	Low	$\mathbf{0.4147 \pm 0.0238}$	$\mathbf{0.6105 \pm 0.0178}$	$\mathbf{0.6069 \pm 0.0205}$
BINALPHA26	Full	0.3636 ± 0.0124	0.6732 ± 0.0258	0.6527 ± 0.0297
	Low	$\mathbf{0.4422 \pm 0.0255}$	$\mathbf{0.6771 \pm 0.0221}$	$\mathbf{0.6578 \pm 0.0281}$
VOWEL	Full	0.2960 ± 0.0405	0.3010 ± 0.0402	0.2960 ± 0.0417
	Low	$\mathbf{0.2980 \pm 0.0323}$	$\mathbf{0.3040 \pm 0.0304}$	$\mathbf{0.3020 \pm 0.0314}$
MNIST	Full	0.4067 ± 0.0830	0.4467 ± 0.1043	0.8067 ± 0.0435
	Low	$\mathbf{0.4400 \pm 0.1020}$	$\mathbf{0.7933 \pm 0.0772}$	$\mathbf{0.8267 \pm 0.0742}$
JAFFE	Full	0.6519 ± 0.1066	0.9446 ± 0.0479	0.9870 ± 0.0188
	Low	$\mathbf{0.8617 \pm 0.0813}$	$\mathbf{1.0000 \pm 0.0000}$	$\mathbf{0.9951 \pm 0.0098}$

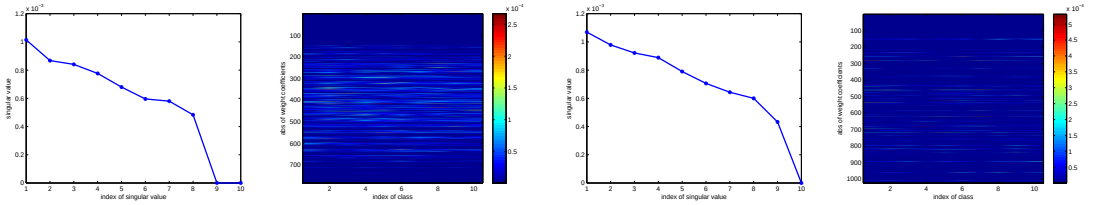
Please note that all the data are centered and we consider the model without bias. The code is written in MATLAB and we terminate our iterative optimization procedure of sparse regression when the relative change in the objective function is below 10^{-5}.

5.4.3 Classification Results

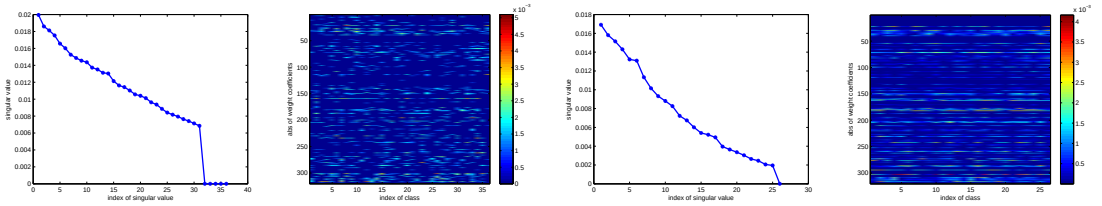
Our proposed methods can find the low-rank structure of the regression models, which are equivalent to doing regression in the regularized LDA subspace. For illustra-



(a) UMIST low-rank structure and sparse structure (b) VOWEL low-rank structure and sparse structure



(c) MNIST low-rank structure and sparse structure (d) JAFFE low-rank structure and sparse structure



(e) BINALPHA36 low-rank structure and sparse structure (f) BINALPHA26 low-rank structure and sparse structure

Figure 5.1. Demonstration of the low-rank structure and sparse structure found by our proposed SLRR method..

tion purpose, in Fig. 5.1 we plot the ranked singular value of the learnt coefficient matrix $W = AB$ on the left hand side and draw the absolute value of the learnt W of the 1st fold (of the 5 fold cross validation, other folds show similar result) on the right hand side for each dataset. The corresponding rank parameter is selected based on which SLRR achieves the best classification accuracy. For example, in Fig. 5.1(a) shows the UMIST results, we can see the number of non-zero singular value of W is 15, i.e., the rank of the learnt coefficient matrix is 15, less than its full rank value of 20. In addition, the learnt W is sparse and is effectively used for feature selection, *e.g.* selecting the important features (non-zero rows) across all the classes. Fig. 5.2 shows the average classification accuracy comparisons of the above three types of full-rank regressions with the proposed low-rank counterparts

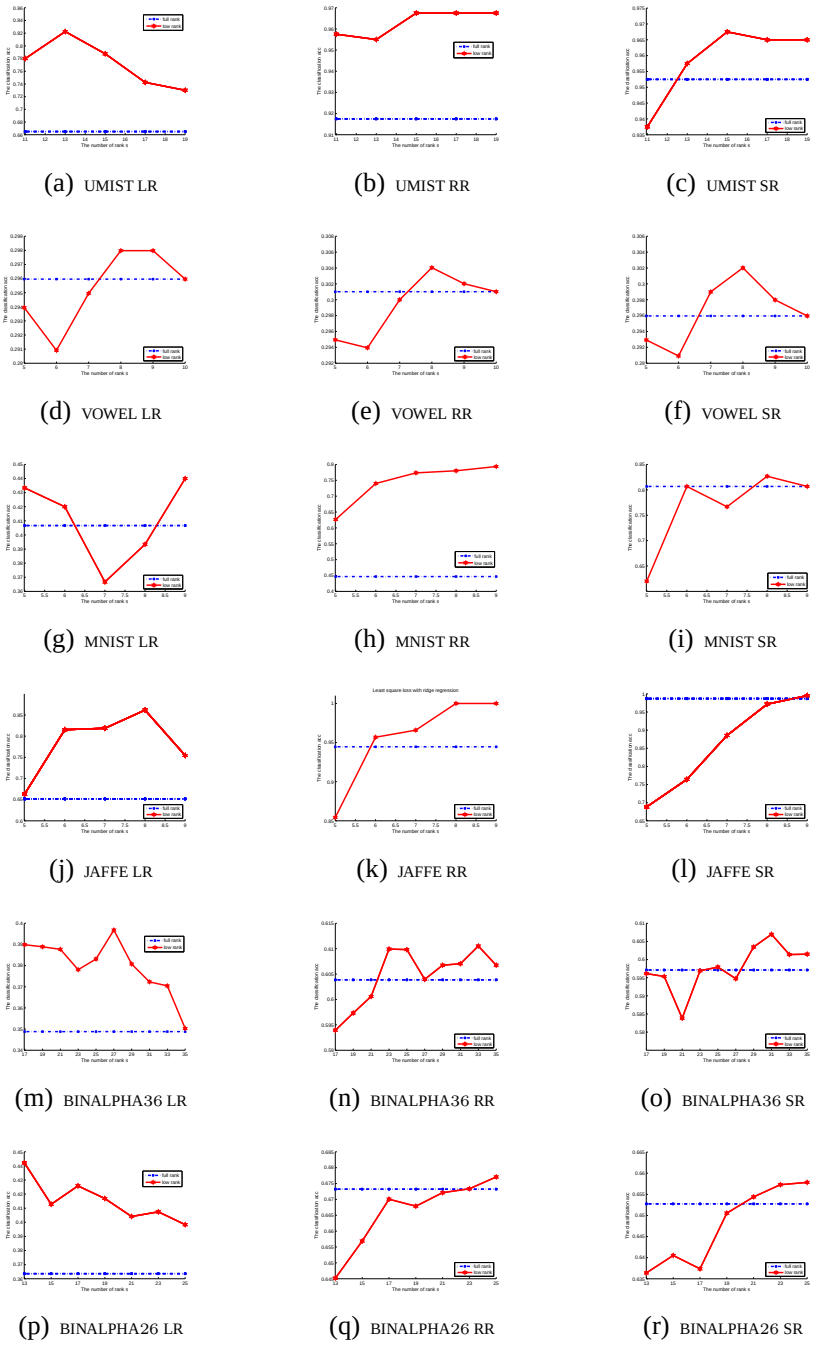
with respect to different low-rank constraints. From Fig. 5.2, we can obviously conclude that the discriminant low-rank regressions consistently outperform their full-rank counterparts, when the specified low-rank parameter s falls in a proper range. For five out of six datasets in our experiments, the low-rank property can boost the result greatly. Only in JAFFE dataset (as shown in Fig. 5.2.(l)), the performance of sparse low-rank regression is competitive with that of the full-rank counterpart.

To help the researchers easily compare all methods, we also list the best classification results in terms of average accuracy and standard deviation for different regression methods in Table 5.2.

Our experimental results also verify our previous key point that the RLRR method is better than LRLR method. On all six datasets, the RLRR outperforms the LRLR. Surprisingly, the standard ridge regression even has better performance than the LRLR method. The LRLR is equivalent to existing low-rank regression models, and both methods may have suboptimal results due to the rank deficiency problem. In standard ridge regression or RLRR methods, because the rank constraint is imposed, both of them alleviate such matrix rank deficiency issue. Now we showed the connection between low-rank constraint and LDA projection, such that we can uncover this problem.

For some data with very large feature dimension ($d >> n$), like UMIST, MNIST and JAFFE, feature selection is necessary to reduce the redundancy between features and alleviate the curse of dimensionality. Our classification results both in Fig. 5.2 and Table 5.2 have shown that under such circumstances, SLRR and its full rank counterpart can achieve better classification result than RLRR and ridge regression since the $\ell_{2,1}$-norm can impose sparsity and select the features for all the classes.

Thus, our newly proposed RLRR as well as SLRR methods are more important and more practical low-rank models for machine learning applications.



(a) UMIST LR (b) UMIST RR (c) UMIST SR

(d) VOWEL LR (e) VOWEL RR (f) VOWEL SR

(g) MNIST LR (h) MNIST RR (i) MNIST SR

(j) JAFFE LR (k) JAFFE RR (l) JAFFE SR

(m) BINALPHA36 LR (n) BINALPHA36 RR (o) BINALPHA36 SR

(p) BINALPHA26 LR (q) BINALPHA26 RR (r) BINALPHA26 SR

Figure 5.2. The average classification accuracy using 5-fold cross validation on six datasets.

5.5 Conclusion

In this chapter, we provide theoretical analysis on low-rank regression models. We proved that the low-rank regression is equivalent to doing linear regression in the LDA subspace. More important, we proposed two new discriminant low-rank ridge regression and sparse low-rank regression methods. Both of them are equivalent to doing regularized regression in the regularized LDA subspace. From both theoretical and empirical views, we showed that both LRRR and SLRR methods provide better learning results than standard low-rank regression. Extensive experiments have been conducted on six benchmark datasets to demonstrate that our proposed low-rank regression methods consistently outperform their corresponding full-rank counterparts in terms of average classification accuracy.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this dissertation, we have proposed several methods to tackle the learning big heterogeneous data problems.

Specifically, if the number of feature or the length of the data descriptor is high, we could use $\ell_{2,1}$-norm SVM to select important features with respect to all the classes. Moreover, if we want to select exact K features and do not want to bother tuning the regularization parameter, we can resort to the proposed feature selection method with $\ell_{2,0}$-norm constraint. Although the latter will find the local solution since the proposed model is not a convex problem, we can always find a good starting point and get a reasonable solution.

If the data is collected from different sources or represented by multiple descriptors, we proposed graph based multi-modality learning models to do either spectral clustering or semi-supervised learning to fuse those heterogeneous information. Moreover, if the data number is huge, we propose the robust multi-view K-Means model to cluster big heterogeneous data without the heavy burden of graph construction.

At last, if the number of classes is large, we give a global solution to low-rank linear regression and prove that the low-rank regression is equivalent to doing linear regression in the corresponding linear discriminant analysis (LDA) space.

6.2 Future Work

In the coming big data era, the number of categories of data can be increased dramatically. When the number of classes becomes large, how to utilize the correlation between

classes to learn heterogeneous multi-modality data becomes the hot topic right now, which can be coped with the learning model with new group lasso and low-rank regularization. In addition, kernel learning can be combined into our proposed methods to handle the non-linear data.

REFERENCES

[1] S. Xiang, F. Nie, and C. Zhang, "Learning a mahalanobis distance metric for data clustering and classification," *Pattern Recognition*, vol. 41, no. 12, pp. 3600–3612, 2008.

[2] G. Forman and E. Kirshenbaum, "Extremely fast text feature extraction for classification and indexing," in *CIKM*, 2008, pp. 1221–1230.

[3] H. Peng, F. Long, and C. H. Q. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005.

[4] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.

[5] H. Liu and H. Motoda, *Feature selection for knowledge discovery and data mining*. Springer, 1998.

[6] J. Habbema and J. Hermans, "Selection of variables in discriminant analysis by F-statistic and error rate," *Technometrics*, vol. 19, no. 4, pp. 487–493, 1977.

[7] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *ML*, 1992, pp. 249–256.

[8] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the gini index and information gain criteria," *Ann. Math. Artif. Intell.*, vol. 41, no. 1, pp. 77–93, 2004.

[9] F. Nie, S. Xiang, Y. Jia, C. Zhang, and S. Yan, "Trace ratio criterion for feature selection," in *AAAI*, 2008.

[10] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, no. 1-2, pp. 273–324, 1997.

[11] M. A. Hall and L. A. Smith, “Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper,” in *FLAIRS Conference*, 1999, pp. 235–239.

[12] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[13] A. Prinzie and D. V. den Poel, “Random forests for multiclass classification: Random multinomial logit,” *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1721–1732, 2008.

[14] P. S. Bradley and O. L. Mangasarian, “Feature selection via concave minimization and support vector machines,” in *ICML*, 1998, pp. 82–90.

[15] L. Wang, J. Zhu, and H. Zou, “Hybrid huberized support vector machines for microarray classification,” in *ICML*, 2007, pp. 983–990.

[16] G. Obozinski, B. Taskar, and M. Jordan, “Multi-task feature selection,” *Statistics Department, UC Berkeley, Tech. Rep*, 2006.

[17] A. Argyriou, C. A. Micchelli, M. Pontil, and Y. Ying, “A spectral regularization framework for multi-task structure learning,” in *NIPS*, 2007.

[18] N. Cristianini and J. Shawe-Taylor, *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge university press, 2004.

[19] F. Nie, H. Huang, X. Cai, and C. H. Q. Ding, “Efficient and robust feature selection via joint ;2, 1-norms minimization,” in *NIPS*, 2010, pp. 1813–1821.

[20] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[21] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, “Liblinear: A library for large linear classification,” *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[22] C. Nutt, D. Mani, R. Betensky, P. Tamayo, J. Cairncross, C. Ladd, U. Pohl, C. Hartmann, M. McLaughlin, T. Batchelor, *et al.*, "Gene expression-based classification of malignant gliomas correlates better with survival than histological classification," *Cancer Research*, vol. 63, no. 7, p. 1602, 2003.

[23] D. Singh, P. Febbo, K. Ross, D. Jackson, J. Manola, C. Ladd, P. Tamayo, A. Renshaw, A. D'Amico, J. Richie, *et al.*, "Gene expression correlates of clinical prostate cancer behavior," *Cancer cell*, vol. 1, no. 2, pp. 203–209, 2002.

[24] S. Fodor, "Massively parallel genomics," *Science(Washington)*, vol. 277, no. 5324, pp. 393–395, 1997.

[25] A. Su, J. Welsh, L. Sapinoso, S. Kern, P. Dimitrov, H. Lapp, P. Schultz, S. Powell, C. Moskaluk, H. Frierson, *et al.*, "Molecular classification of human carcinomas by use of gene expression signatures," *Cancer Research*, vol. 61, no. 20, p. 7388, 2001.

[26] S. Armstrong, J. Staunton, L. Silverman, R. Pieters, M. den Boer, M. Minden, S. Sallan, E. Lander, T. Golub, and S. Korsmeyer, "MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia," *Nature genetics*, vol. 30, no. 1, pp. 41–47, 2001.

[27] I. Kononenko, "Estimating attributes: Analysis and extensions of relief," in *ECML*, 1994, pp. 171–182.

[28] X. Zhou and D. P. Tuck, "Msvm-rfe: extensions of svm-rfe for multiclass gene selection on dna microarray data," *Bioinformatics*, vol. 23, no. 9, pp. 1106–1114, 2007.

[29] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient l2, 1-norm minimization," in *UAI*, 2009, pp. 339–348.

[30] X. Cai, F. Nie, H. Huang, and C. H. Q. Ding, "Multi-class $\ell_{2,1}$-norms support vector machine," in *ICDM*, 2011, pp. 91–100.

[31] L. Mancera and J. Portilla, "L0-norm-based sparse representation through alternate projections," in *ICIP*, 2006, pp. 2089–2092.

[32] G. Obozinski, B. Taskar, and M. I. Jordan, "Joint covariate selection and joint subspace selection for multiple classification problems," *Statistics and Computing*, vol. 20, no. 2, pp. 231–252, 2010.

[33] C. H. Q. Ding, D. Zhou, X. He, and H. Zha, "R_1-pca: rotational invariant l_1-norm principal component analysis for robust subspace factorization," in *ICML*, 2006, pp. 281–288.

[34] D. Bertsekas, "Constrained optimization and lagrange multiplier methods," *Computer Science and Applied Mathematics, Boston: Academic Press, 1982*, vol. 1, 1982.

[35] D. P. Bertsekas, *Constrained optimization and lagrange multiplier methods*. Athena Scientific, 1996.

[36] M. J. D. Powell, *A method for nonlinear constraints in minimization problems*. In R. Fletcher, editor, Optimization. Academic Press, London and New York, 1969.

[37] R. Duda, P. Hart, and D. Stork, "Pattern classification and scene analysis 2nd ed." 1995.

[38] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *NIPS*, 2006, pp. 41–48.

[39] D. Luo, C. H. Q. Ding, and H. Huang, "Towards structural sparsity: An explicit l2/l0 approach," in *ICDM*, 2010, pp. 344–353.

[40] A. Biswas and D. Jacobs, "Active Image Clustering: Seeking Constraints from Humans to Complement Algorithms," *CVPR*, pp. 2152–2159, 2012.

[41] Y. J. Lee and K. Grauman, "Foreground focus: Unsupervised learning from partially matching images," *International Journal of Computer Vision*, vol. 85, no. 2, pp. 143–166, 2009.

[42] D. Dueck and B. J. Frey, "Non-metric affinity propagation for unsupervised image categorization," in *ICCV*, 2007, pp. 1–8.

[43] X. Cai, F. Nie, H. Huang, and F. Kamangar, "Heterogeneous image features integration via multi-modal spectral clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011)*, 2011, pp. 1977–1984.

[44] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[45] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR (1)*, 2005, pp. 886–893.

[46] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, 2002.

[47] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[48] J. Wu and J. M. Rehg, "Where am i: Place instance and category recognition using spatial pact," in *CVPR*, 2008.

[49] H. Yu, M. Li, H. Zhang, and J. Feng, "Color texture moments for content-based image retrieval," in *ICIP (3)*, 2002, pp. 929–932.

[50] C. H. Q. Ding, X. He, and H. D. Simon, "Nonnegative lagrangian relaxation of *k*-means and spectral clustering," in *ECML*, 2005, pp. 530–538.

[51] D. Yan, L. Huang, and M. I. Jordan, "Fast approximate spectral clustering," in *KDD*, 2009, pp. 907–916.

[52] T. Sakai and A. Imiya, "Fast spectral clustering with random projection and sampling," in *MLDM*, 2009, pp. 372–384.

[53] M. F. Duarte and Y. H. Hu, "Vehicle classification in distributed sensor networks," *J. Parallel Distrib. Comput.*, vol. 64, no. 7, pp. 826–838, 2004.

[54] F.-F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[55] J. M. Winn and N. Jojic, "Locus: Learning object classes with unsupervised segmentation," in *ICCV*, 2005, pp. 756–763.

[56] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[57] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *CVPR*, 2009, pp. 951–958.

[58] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *CVPR*, 2010, pp. 3485–3492.

[59] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM TIST*, vol. 2, no. 3, p. 27, 2011.

[60] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *CVPR*, 2007.

[61] A. Bosch, A. Zisserman, and X. Muñoz, "Representing shape with a spatial pyramid kernel," in *CIVR*, 2007, pp. 401–408.

[62] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluation of color descriptors for object and scene recognition," in *CVPR*, 2008.

[63] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[64] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[65] L. Cao, J. Luo, F. Liang, and T. Huang, "Heterogeneous feature machines for visual recognition," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2010, pp. 1095–1102.

[66] R. Fergus, F.-F. L. 0002, P. Perona, and A. Zisserman, "Learning object categories from google's image search," in *ICCV*, 2005, pp. 1816–1823.

[67] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman, "Discovering objects and their localization in images," in *ICCV*, 2005, pp. 370–377.

[68] T. Hofmann, "Unsupervised learning by probabilistic latent semantic analysis," *Machine Learning*, vol. 42, no. 1/2, pp. 177–196, 2001.

[69] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[70] K. Grauman and T. Darrell, "Unsupervised learning of categories from sets of partially matching image features," in *CVPR (1)*, 2006, pp. 19–25.

[71] ——, "The pyramid match kernel: Discriminative classification with sets of image features," in *ICCV*, 2005, pp. 1458–1465.

[72] F.-F. L. 0002, P. Perona, and C. I. of Technology, "A bayesian hierarchical model for learning natural scene categories," in *CVPR (2)*, 2005, pp. 524–531.

[73] X. Wang, T. Han, and S. Yan, "An HOG-LBP human detector with partial occlusion handling," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2010, pp. 32–39.

[74]

[75] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *NIPS*, 2001, pp. 849–856.

[76] C. Ding, T. Li, and M. Jordan, "Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding," in *Data Min-*

ing, 2008. ICDM'08. Eighth IEEE International Conference on. IEEE, 2009, pp. 183–192.

[77] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, 2001.

[78] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *NIPS*, 2004.

[79] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, 2003, pp. 912–919.

[80] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *NIPS*, 2003.

[81] D. Zhou and B. Schölkopf, "Learning from labeled and unlabeled data using random walks," in *DAGM-Symposium*, 2004, pp. 237–244.

[82] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

[83] C. H. Q. Ding, R. Jin, T. Li, and H. D. Simon, "A learning framework using green's function and kernel regularization with application to recommender system," in *KDD*, 2007, pp. 260–269.

[84] D. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality," *AMS Math Challenges Lecture*, pp. 1–32, 2000.

[85] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic Pr, 1990.

[86] T. Anderson, "Estimating linear restrictions on regression coefficients for multivariate normal distributions," *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 327–351, 1951.

[87] S. Xiang, Y. Zhu, X. Shen, and J. Ye, "Optimal exact least squares rank minimization," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 480–488.

[88] F. Bunea, Y. She, and M. Wegkamp, "Optimal selection of reduced rank estimators of high-dimensional matrices," *The Annals of Statistics*, vol. 39, no. 2, pp. 1282–1309, 2011.

[89] A. Izenman, "Reduced-rank regression for the multivariate linear model," *Journal of multivariate analysis*, vol. 5, no. 2, pp. 248–264, 1975.

[90] T. Anderson, "Asymptotic distribution of the reduced rank regression estimator under general conditions," *The Annals of Statistics*, vol. 27, no. 4, pp. 1141–1154, 1999.

[91] G. Reinsel and R. Velu, *Multivariate reduced-rank regression: theory and applications*. Springer New York, 1998.

[92] J. Ye, "Least squares linear discriminant analysis," in *ICML*, 2007, pp. 1087–1093.

[93] D. Luo, C. Ding, and H. Huang, "Linear discriminant analysis: New formulations and overfit analysis," *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[94] A. Hoerl and R. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, pp. 55–67, 1970.

[95] J. Friedman, "Regularized discriminant analysis," *Journal of the American statistical association*, pp. 165–175, 1989.

[96] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[97] P. Zhao and B. Yu, "On model selection consistency of lasso," *Journal of Machine Learning Research*, vol. 7, pp. 2541–2563, 2006.

[98] L. Sun, R. Patel, J. Liu, K. Chen, T. Wu, J. Li, E. Reiman, and J. Ye, "Mining brain region connectivity for alzheimer's disease study via sparse inverse covariance estimation," in *KDD*, 2009, pp. 1335–1344.

[99] H. Wang, F. Nie, H. Huang, S. L. Risacher, C. Ding, A. J. Saykin, L. Shen, and ADNI, "A new sparse multi-task regression and feature selection method to identify

brain imaging predictors for memory performance," *IEEE Conference on Computer Vision*, pp. 557–562, 2011.

[100] H. Wang, F. Nie, H. Huang, J. Yan, S. Kim, S. Risacher, A. Saykin, and L. Shen, "High-order multi-task feature learning to identify longitudinal phenotypic markers for alzheimer's disease progression prediction," in *NIPS*, 2012, pp. 1286–1294.

[101] H. Wang, F. Nie, H. Huang, *et al.*, "Identifying disease sensitive and quantitative trait-relevant biomarkers from multidimensional heterogeneous imaging genetics data via sparse multimodal multitask learning," *Bioinformatics*, vol. 28, no. 12, pp. i127–i136, 2012.

[102] D. Graham and N. Allinson, "Characterising virtual eigensignatures for general purpose face recognition," *NATO ASI series. Series F: computer and system sciences*, pp. 446–456, 1998.

[103] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 19, no. 7, pp. 711–720, 1997.

[104] M. Niranjan and F. Fallside, "Neural networks and radial basis functions in classifying static speech patterns," *Computer Speech & Language*, vol. 4, no. 3, pp. 275–289, 1990.

[105] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[106] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets," in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. IEEE, 1998, pp. 200–205.

BIOGRAPHICAL STATEMENT

Xiao Cai was born in Beijing, P.R. China, in 1983. He received his B.S. degree from Tianjin University, P.R. China, in 2007. And he got his M.S. degree focusing on image processing and signal processing from New Jersey Institute of Technology in 2009. Since 2009, he joined Computational Science Lab as a graduate teaching and research assistant. He received his Ph.D. degree from the University of Texas at Arlington in 2013 in Computer Engineering. He did internship in Abbott Laboratories in 2012. His current research interest is in the area of machine learning, data mining, computer vision and medical image analysis. He is a student member of IEEE.